

**Publication number: JP2001515242 (T)**

**Inventor(s):**

**Applicant(s):**

**Classification:**

- international: **G06F9/46; G06F9/48; G06F9/46; (IPC1-7): G06F9/46**

- European: G06F9/48C4S

**Application number: JP20000509034T 19980903**

**Priority number(s):** US19970923173 19970904; WO1998US18530 19980903

JP4174178 (B2)  
WO9912097 (A1)  
US6385638 (B1)  
TW439031 (B)  
EP1010074 (A1)

more >>

Abstract of corresponding document: **WO 9912097 (A1)**

```

graph TD
    User[1 User] -- "User starts applications" --> UserMonitor[User monitor with own policy preferences]
    UserMonitor --> PolicyBox[0 Policy Box]
    PolicyBox -- "Use Policy Box if system is overloaded" --> AdmissionControl[6 Admission Control]
    PolicyBox --> GrantControl[7 Grant Control]
    AdmissionControl --> ResourceManager[5 Resource Manager]
    ResourceManager -- "Notify scheduler to release or hold grants" --> Scheduler[4 Scheduler]
    Scheduler -- "Release CPU schedule of tasks with grants" --> Processor[3 Processor]
    GrantControl -- "Grant executed via tasks through the Resource Manager" --> Processor
    Applications[2 Applications Tasks] -- "Task requests submitted" --> UserMonitor
    UserMonitor -- "Task requests submitted" --> ResourceManager
  
```

**Scheduling Components in the Processor Resource Distributor**

<http://v3.espacenet.com/publicationDetails/biblio?DB=EPODOC&adjacent=true&loc...> 2010/10/22

(11)特許出願公表番号  
特表2001-515242  
(P2001-515242A)

(43)公表日 平成13年9月18日(2001.9.18)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	データベース* (参考)
G 0 6 F 9/46	3 4 0	G 0 6 F 9/46	3 4 0 E 5 B 0 9 8 3 4 0 B

審查請求 未請求 予備審查請求 有 (全 57 頁)

(21) 出願番号	特願2000-509034(P2000-509034)
(86) (22) 出願日	平成10年9月3日(1998.9.3)
(85) 翻訳文提出日	平成12年3月3日(2000.3.3)
(86) 国際出願番号	PCT/US98/18530
(87) 国際公開番号	WO99/12097
(87) 国際公開日	平成11年3月11日(1999.3.11)
(31) 優先権主張番号	08/923, 173
(32) 優先日	平成9年9月4日(1997.9.4)
(33) 優先権主張国	米国(US)

(71)出願人 イクエーター テクノロジーズ インコーポレイテッド  
アメリカ合衆国 98101-4001 ワシントン州 シアトル 520 パイク ストリート スウィート 900

(72)発明者 ベイカー・ハーヴェイ、 ミシュ  
アメリカ合衆国 98112 ワシントン州 シアトル セブンティーンズ アヴェニュー イースト 1212

(74)代理人 弁理士 松永 宣行

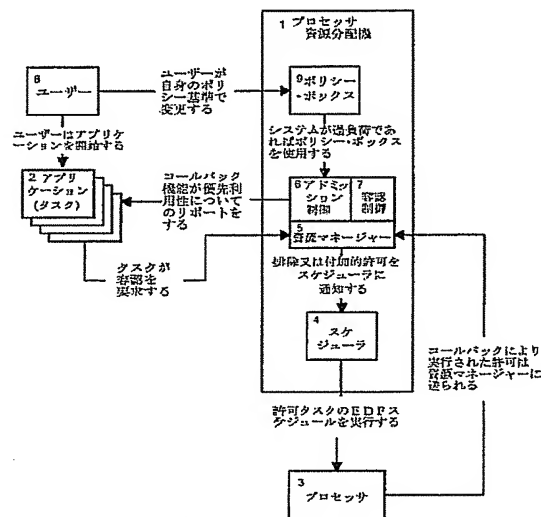
[最終頁に続く](#)

(54) 【発明の名称】 プロセッサ資源分配装置及び方法

(57) 【要約】

【目的】 プロセッサの資源を有効に利用すること

【解決手段】 方法及び装置は、2以上の択一的タスク又はスレッドの中でプロセッサに得られる時間を分配し、それらのスケジューリングをする。各タスクは、作動可能の基でプロセスの利用時間に関し、最小時間を使用する低品質のサービスと、最高時間を使用する最良品質サービスとを含むマルチレベルの特徴を有する。方法及び装置は、各タスクが低品質のサービスで作動することができることを保証し、全てのサービスから最適な品質のサービスを得るための方法を含む。



プロセッサ資源分配機のスケジューリング要素

**【特許請求の範囲】**

**【請求項1】** コンピュータ・システムにおいてプロセッサで実行されるべきタスクを容認するための方法であって、

(a) プロセッサの実行要求を少なくとも3つのタスクから受ける工程であって、高品質のパフォーマンスを提供する少なくとも高いレベルのプロセッサ使用時間をそれぞれが有し、少なくとも2つのタスクが低品質のパフォーマンスを提供する低いレベルのプロセッサ使用時間を必要とする付加的な低使用レベルを有する、工程と、

(b) 付加的な低使用レベルの少なくとも1つのタスクを含む少なくとも2つのタスクであるが全てのタスクではないタスクを処理するために容認する工程であって、前記プロセッサの未使用時間が低使用レベルの残存タスクの1つを容認するに不十分であり、低使用レベルの和が前記プロセッサに得ることができるトータル時間を超えないように、容認タスクを選択する工程と、

(c) 前記残存タスクを処理するための容認から排除する工程とを含む、方法。

**【請求項2】** 前記プロセッサの時間が許可された全てのタスクを高使用レベルで実行させるのに不十分であるとき、さらに、

(d) 低使用レベルで実行する容認タスクの少なくとも1つで前記容認タスクの実行を開始する工程と、

(e) 1以上の容認タスクが実行の要求を一時的に中止しかつ前記プロセッサが低使用レベルで実行しているタスクを高使用レベルで実行させるに十分な未使用時間を得ることができるとき、前記タスクを高使用レベルに切り換える工程とを含む、請求項1に記載の方法。

**【請求項3】** さらに、(d)容認タスクの実行を開始する工程と、

(e) 1以上の容認タスクが実行要求を一時的に中止しかつ前記プロセッサが実行を要求している容認されなかったタスクを実行させるに十分な未使用時間を得ることができるとき、容認されなかったタスクを容認から排除する工程とを含む、請求項1に記載の方法。

**【請求項4】** さらに、(d)前記容認タスクの実行を開始する工程と、

(e) 以前に排除されたタスクが容認を再度要求しているとき、以前の容認タスクの1つが終了しないかぎり及び残りの容認タスクの低使用レベルの和が前記プロセッサに得ることができる時間を超えることなく以前の排除タスクを容認するのに十分に小さくならないかぎり、前記タスクを再度排除する工程とを含む、請求項1に記載の方法。

【請求項5】 少なくとも1つの容認タスクは少なくとも3つの使用レベルを有する、請求項1に記載の方法。

【請求項6】 コンピュータ・システムにおいてプロセッサ時間をタスクに割り当てるべくプロセッサ資源分配機のための方法であって、

(a) プロセッサ時間の要求を2以上のタスクから受ける工程であって、少なくとも1つの要求は前記要求と関連するバイレベル・タスク(bi-level task)のために2以上のプロセッサ使用レベルの仕様を含み、前記2以上のレベルはリアル・タイムの分配のためのサイクル持続期間により特定され、各サイクル期間のためには前記バイレベル・タスクに配分すべき各サイクルの一部の仕様を含む工程と、

(b) 前記要求に基づいて前記2以上のタスクのそれぞれにプロセッサ時間を配分する工程であって、各サイクルの一部及び前記特定されたサイクル持続期間の1つを選択することにより前記バイレベル・タスクのためのプロセッサ時間を配分する工程とを含む、方法。

【請求項7】 前記2以上のレベルのそれぞれは前記バイレベル・タスクのためのコールバック機能と関連し、前記タスクは選択されかつ特定されたサイクル持続期間及び各サイクルの一部と関連するコールバック機能を使用することにより呼び出される、請求項6に記載の方法。

【請求項8】 前記選択は得られるプロセッサ時間量を基に行われ、選択されたレベルは得られるプロセッサ時間量内に調和し得る各サイクルの最大位置と関連するレベルである、請求項6に記載の方法。

【請求項9】 コンピュータ・システムにおいて少なくとも2つのタスク間でプロセッサの時間配分を調整するための方法であって、

(a) 前記プロセッサの時間を一連の第1の期間に分割し、第1のタスクの前

記プロセッサでの実行を開始し、各第1の期間の一部を前記第1のタスクに配分する工程と、

(b) 前記プロセッサ時間を前記一連の第1の期間と同じであるか又は異なる一連の第2の期間に分割し、高品質のパフォーマンスを提供するレベルであって各第2の期間を高レベルの使用で実行する第2のタスクの前記プロセッサでの実行を開始し、各第2の期間の一部を第2のタスクに配分する工程であって、前記第2のタスクは前記高レベルに比べ低いプロセッサ時間を使用する低いレベルであって前記高レベルに比べ低品質のパフォーマンスを提供する低レベルの使用でも各第2の期間を実行することができる、工程と、

(c) 第1のタスクが配分されたプロセッサ時間に比べ期間において大きいプロセッサ時間を供給しているとき、第2のタスクを低レベル使用に切り換え、第1のタスクで利用可能にすべくプロセッサの時間の少なくとも一部を再配分する工程とを含む、方法。

【請求項10】 第2のタスクの低レベル仕様への切り換えは、一連の第2の期間のそのときの各期間の終わりに発生し、引き続く期間のために効果を与える、請求項9に記載の方法。

【請求項11】 第1のタスクが配分時間よりも期間において大きいプロセッサ時間を供給しているか否かの決定はタスクが前記期間に要求している時間の推定を基にした予測により行われ、切り換え及び再配分は第1のタスクが前記期間の配分時間を消費する前に行われる、請求項9に記載の方法。

【請求項12】 第2のタスクの低レベル使用への切り換えは、前記一連の第2の期間のそのときの期間の終わりに発生し、引き続く期間のために効果をもたらす、請求項11に記載の方法。

【請求項13】 さらに、(d) 工程(b)及び(c)間において、高品質のパフォーマンスを提供するレベルを有しかつ高レベルのプロセッサ使用時間を実行する第3のタスクのプロセッサでの実行を開始する工程であって、第3のタスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる、工程と、

(e) 工程(c)において、第1のタスクが配分されたプロセッサ時間より大きい時間を要求している場合に、第3のタスクよりもむしろ第2のタスクを低い使用レベルに切り換える先に作られた選択を基に、第3のタスクよりもむしろ第2のタスクを低レベルの使用に切り換える工程とを含む、請求項9に記載の方法。

【請求項14】 コンピュータ・システムにおいてプロセッサの使用時間をタスクにより調整する方法であって、

(a) プロセッサの時間を一連の期間に分割する工程と、

(b) 高品質のパフォーマンスを提供する高いレベルのプロセッサ使用時間で実行するタスクのプロセッサでの実行を開始する工程であって、前記タスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低い品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる、工程と、

(c) 前記各期間の一部をタスクに配分する工程と、

(d) 前記タスクが配分されたプロセッサ時間より多いプロセッサ時間を要求しているとき、そのタスクを低使用レベルに切り換える工程とを含む、方法。

【請求項15】 前記タスクは次の期間のために低使用レベルに切り替えられる、請求項14に記載の方法。

【請求項16】 前記タスクが配分されたプロセッサ時間より期間において大きいプロセッサ時間を要求していることの決定は前記タスクが前記期間において要求している時間の推定に基づく予測により行われ、前記切り換えは前記タスクが前記期間において配分された時間を消費する前に行われる、請求項14に記載の方法。

【請求項17】 コンピュータ・システムにおいてプロセッサの使用時間をタスクにより調整する方法であって、

(a) プロセッサの時間を一連の期間に分割する工程と、

(b) 高品質のパフォーマンスを提供する高レベルのプロセッサ使用時間で実行する第1のタスクのプロセッサでの実行を開始する工程であって、前記タスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベル

より低品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる、工程と、

(c) 前記期間の一部を第1のタスクに配分する工程と、

(d) 高品質のパフォーマンスを提供する高いレベルのプロセッサ使用時間で実行する第2のタスクのプロセッサでの実行を開始する工程であって、前記第2のタスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる工程と、

(e) 第1のタスクが配分されたプロセッサ時間より期間において大きいプロセッサ時間を要求しているとき、第1のタスクが配分されたプロセッサ時間より多いプロセッサ時間を要求している場合に、第1のタスクよりもむしろ第2のタスクを低レベルの使用に切り換える先に作られた選択を基に、第1のタスクを低レベル使用に切り換えるよりもむしろ第2のタスクを低レベルの使用に切り換える工程とを含む、方法。

【請求項18】 コンピュータシステムにおいて作動されたとき、プロセッサで実行されるべき容認タスクのための以下の方法をシステムに実行させるコンピュータ・プログラムを含む、コンピュータ読み取り可能な媒体であって、

(a) 少なくとも3つのタスクから前記プロセッサの実行要求を受ける工程であって、高品質のパフォーマンスを提供する少なくとも高いレベルのプロセッサ使用時間をそれぞれが有し、少なくとも2つのタスクは低品質のパフォーマンスを提供する低いレベルのプロセッサ使用時間を要求する付加的な使用レベルを有する、工程と、

(b) 付加的な低使用レベルの少なくとも1つのタスクを含む少なくとも2つのタスクであるが全てのタスクではないタスクを処理するために容認する工程であって、プロセッサの未使用時間が低使用レベルの残存タスクの1つを容認するに不十分であり、低使用レベルの和が前記プロセッサに得られるトータル時間を超えないように、容認するタスクを選択する工程と、

(c) 残存タスクを処理するために容認から排除する工程とを含む、コンピュータ読み取り可能な媒体。

【請求項19】 さらに、前記プロセッサの時間が許可された全てのタスクを高使用レベルで実行させるのに不十分であるときに、前記方法は、さらに、

(d) 低使用レベルで実行する容認タスクの少なくとも1つで前記容認タスクの実行を開始する工程と、

(e) 1以上の容認タスクが実行の要求を一時的に中止しかつ前記プロセッサが低使用レベルで実行しているタスクを高使用レベルで実行させるに十分な未使用時間を得ることができるとき、前記タスクを高使用レベルに切り換える工程を含む、請求項18に記載のコンピュータ読み取り可能な媒体。

【請求項20】 前記方法は、さらに、(d) 容認タスクの実行を開始する工程と、

(e) 1以上の容認タスクが実行の要求を一時的に中止しかつ前記プロセッサが実行を要求している容認されなかったタスクを実行させるに十分な未使用時間を得ることができるとき、容認されなかったタスクを容認から排除する工程を含む、請求項18に記載のコンピュータ読み取り可能な媒体。

【請求項21】 前記方法は、さらに、(d) 容認タスクの実行を開始する工程と、

(e) 以前に排除されたタスクが容認をさらに要求しているとき、前に容認したタスクの1つが終了しないかぎり及び残存許可タスクの低レベル使用の和がプロセッサの時間を越えることなく前に実行したタスクを許可することができる程度に十分小さくならないかぎり、そのタスクをさらに排除する工程を含む、請求項18に記載のコンピュータ読み取り可能な媒体。

【請求項22】 前記少なくとも1つのタスクは少なくとも3つの使用レベルを有する、請求項18に記載のコンピュータ読み取り可能な媒体。

【請求項23】 コンピュータシステムにおいて作動されたとき、プロセッサ時間をタスクに割り当てるべくプロセッサ資源分配機のための以下の方法をシステムに実行させるコンピュータプログラムを含む、コンピュータ読み取り可能な媒体であって、

(a) プロセッサ時間のための要求を2以上のタスクの各々から受ける工程であって、少なくとも1つの要求は前記要求と関係するバイレベル・タスクのため



の2以上のプロセッサ使用時間の仕様を含み、2以上のプロセッサ仕様レベルのそれぞれはリアル・タイム分割のためのサイクル持続期間により特定され、及び、各サイクル持続期間のためには前記バイレベル・タスクに配分されるべき各サイクルの一部の仕様を含む、工程と、

(b) 前記要求を基に2以上のタスクのそれぞれにプロセッサ時間を配分し、各サイクルの一部及び特定されたサイクル持続期間の1つを選択することにより前記バイレベル・タスクのためのプロセッサ時間を配分する工程とを含む、コンピュータ読み取り可能な媒体。

【請求項24】 2以上のレベルのそれぞれは前記バイレベル・タスクのためのコールバック機能と関連し、前記タスクは各サイクルの一部及び前記選択されかつ特定されたサイクル持続時間と関連するコールバック機能を使用することにより呼び出される、請求項23に記載のコンピュータ読み取り可能な媒体。

【請求項25】 前記選択は得ることができるプロセッサ時間量を基に行われ、選択されたレベルは得ることができるプロセッサ時間量内に調和され得る各サイクルの最大部分と関連するレベルである、請求項23に記載のコンピュータ読み取り可能な媒体。

【請求項26】 コンピュータシステムにおいて作動されたとき、少なくとも2つのタスクの間でプロセッサの時間配分を調整するための以下の方法をシステムに実行させるコンピュータプログラムを含む、コンピュータ読み取り可能な媒体であって、

(a) 前記プロセッサの時間を一連の第1の期間に分割し、第1のタスクの前記プロセッサでの実行を開始し、各第1の期間の一部を第1のタスクに配分する工程と、

(b) 前記プロセッサ時間を前記一連の第1の期間と同じであるか又は異なる一連の第2の期間に分割し、高品質のパフォーマンスを提供するレベルであって各第2の期間を高レベルの使用で実行する第2のタスクの前記プロセッサでの実行を開始し、各第2の期間の一部を第2のタスクに配分する工程であって、前記第2のタスクは前記高レベルに比べ低いプロセッサ時間を使用する低レベルであって前記高レベルに比べ低品質のパフォーマンスを提供する低いレベルの使用で

も各第2の期間を実行することができる、工程と、

(c) 第1のタスクが配分されたプロセッサ時間に比べ期間において大きいプロセッサ時間を供給しているとき、第2のタスクを低レベル使用に切り換え、第1のタスクで利用可能にすべくプロセッサの時間の少なくとも一部を再配分する工程とを含む、コンピュータ読み取り可能な媒体。

【請求項27】 第2のタスクの低レベル仕様への切り換えは、一連の第2の期間のそのときの各期間の終わりに発生し、引き続く期間のために効果を与える、請求項26に記載のコンピュータ読み取り可能な媒体

【請求項28】 第1のタスクが配分時間よりも期間において多いプロセッサ時間を供給しているか否かの決定はタスクが前記期間に要求している時間の推定を基にした予測により行われ、切り換え及び再配分は第1のタスクが前記期間の配分時間を消費する前に行われる、請求項26に記載のコンピュータ読み取り可能な媒体。

【請求項29】 第2のタスクの低レベル使用への切り換えは、前記一連の第2の期間のそのときの期間の終わりに発生し、引き続く期間のために効果をもたらす、請求項28に記載のコンピュータ読み取り可能な媒体。

【請求項30】 前記方法は、さらに、

(d) 工程(b)及び(c)間において、高品質のパフォーマンスを提供するレベルを有しかつ高レベルのプロセッサ使用時間を実行する第3のタスクのプロセッサでの実行を開始する工程であって、第3のタスクは前記高レベルよりも低品質のプロセッサパフォーマンスを提供するレベルであって前記高レベルに比べ低レベルのプロセッサ時間を使用して前記高レベルよりも低レベルのプロセッサ使用時間でも実行することができる、工程と、

(e) 工程(c)において、第1のタスクが配分されたプロセッサ時間よりも多い時間を要求している場合に、第2のタスクが第2のタスクより低い使用レベルに切り替えられる先に作られた選択を基に、第3のタスクを低レベルの使用に切り替える工程とを含む、請求項26に記載のコンピュータ読み取り可能な媒体。

【請求項31】 コンピュータシステムにおいて作動されたとき、プロセッ

サの使用時間をタスクにより調整するための以下の方法をシステムに実行させるコンピュータプログラムを含む、コンピュータ読み取り可能な媒体であって、

(a) プロセッサの時間を一連の期間に分割する工程と、

(b) 高品質のパフォーマンスを提供する高レベルのプロセッサ使用時間で実行するタスクのプロセッサでの実行を開始する工程であって、前記タスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる、工程と、

(c) 前記期間の一部を第1のタスクに配分する工程と、

(d) 前記タスクが配分されたプロセッサ時間より期間において大きいプロセッサ時間を要求しているとき、そのタスクを低使用レベルに切り換える工程とを含む、コンピュータ読み取り可能な媒体。

【請求項32】 前記タスクは次の期間のために低使用レベルに切り替えられる、請求項31に記載のコンピュータ読み取り可能な媒体。

【請求項33】 前記タスクが配分されたプロセッサ時間より期間において大きいプロセッサ時間を要求していることの決定は前記タスクが前記期間において要求している時間の推定に基づく予測により行われ、前記切り換えは前記タスクが前記期間において配分された時間を消費する前に行われる、請求項31に記載のコンピュータ読み取り可能な媒体。

【請求項34】 コンピュータシステムにおいて作動されたとき、プロセッサの使用時間をタスクにより調整するための以下の方法をシステムに実行させるコンピュータプログラムを含む、コンピュータ読み取り可能な媒体であって、

(a) プロセッサの時間を一連の期間に分割する工程と、

(b) 高レベルより高いプロセッサ時間を使用するレベルであって高品質のパフォーマンスを提供する高レベルのプロセッサ使用時間で実行する第1のタスクのプロセッサでの実行を開始する工程であって、前記タスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる、工程と、

(c) 前記悪期間の一部を第1のタスクに配分する工程と、

(d) 高品質のパフォーマンスを提供する高いレベルのプロセッサ使用時間で実行する第2のタスクのプロセッサでの実行を開始する工程であって、前記第2のタスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低い品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる、工程と、

(e) 第1のタスクが配分されたプロセッサ時間より期間において大きいプロセッサ時間を要求しているとき、先に作られた選択を基に第1のタスクを低レベル使用に切り換えるよりも第2のタスクを期間においてより大きいプロセッサ時間に切り換える工程であって、第1のタスクが配分されたプロセッサ時間より大きいプロセッサ時間を要求している場合に、第2のタスクが第1のタスクよりもむしろ低レベルの使用に切り替えられる工程とを含む、コンピュータ読み取り可能な媒体。

【請求項35】 プロセッサで実行されるべきタスクを許可するコンピュータ装置であって、

(a) プロセッサの実行要求を少なくとも3つのタスクから受ける手段であって、高品質のパフォーマンスを提供する少なくとも高いレベルのプロセッサ使用時間をそれぞれが有し、少なくとも2つのタスクが低品質のパフォーマンスを提供する低いレベルのプロセッサ使用時間を必要とする付加的な低使用レベルを有する手段と、

(b) 付加的な低使用レベルの少なくとも1つのタスクを含む少なくとも2つのタスクであるが全てのタスクではないタスクを処理するために容認する工程であって、前記プロセッサの未使用時間が低使用レベルの残存タスクの1つを容認するに不十分であり、低使用レベルの和が前記プロセッサに得ることができるトータル時間を超えないように、容認タスクを選択する手段と、

(c) 前記残存タスクを処理するための容認から排除する手段とを含む、コンピュータ装置。

【請求項36】 さらに、

(d) 低使用レベルで実行する容認タスクの少なくとも1つで容認タスクの実

行を開始する手段と、

(e) 1以上の容認タスクが実行の要求を一時的に中止しかつ前記プロセッサが低使用レベルで実行しているタスクを高使用レベルで実行させるに十分な未使用時間を得ることができるとき、前記タスクを高使用レベルに切り換える手段とを含む、請求項35に記載の装置。

【請求項37】 さらに、(d)容認タスクの実行を開始する手段と、

(e) 1以上の容認タスクが実行要求を一時的に中止しかつ前記プロセッサが実行を要求している容認されなかったタスクを実行させるに十分な未使用時間を得ることができるとき、容認されなかったタスクを容認から排除する手段とを含む、請求項35に記載の装置。

【請求項38】 さらに、前記容認タスクの実行を開始する手段であって、

(e) 以前に排除されたタスクが容認を再度要求しているとき、以前の容認タスクの1つが終了しなければ及び残りの容認タスクの低使用レベルの和が前記プロセッサに得ることができる時間を超えることなく異残の排除タスクを容認するのに十分に小さくならなければ、前記容認タスクを再度排除する手段を含む、請求項1に記載の装置。

【請求項39】 少なくとも1つの容認タスクは少なくとも3つの使用レベルを有する、請求項35に記載のコンピュータ装置。

【請求項40】 コンピュータ・システムにおいてプロセッサ時間をタスクに割り当てるプロセッサ資源分配機であって、

(a) プロセッサ時間の要求を2以上のタスクから受ける手段であって、少なくとも1つの要求は前記要求と関連するバイレベル・タスク(bi-level task)のために2以上のプロセッサ使用レベルの仕様を含み、前記2以上のレベルはリアル・タイムの分配のためのサイクル持続期間により特定され、各サイクル期間のためには前記バイレベル・タスクに配分すべき各サイクルの一部の仕様を含む手段と、

(b) 前記要求に基づいて前記2以上のタスクのそれぞれにプロセッサ時間を配分する手段であって、各サイクルの一部及び前記特定されたサイクル持続期間の1つを選択することにより前記バイレベル・タスクのためのプロセッサ時間を

配分する手段とを含む、プロセッサ資源分配機。

【請求項41】 前記2以上のレベルのそれぞれは前記バイレベル・タスクのためのコールバック機能と関連し、前記タスクは選択されかつ特定されたサイクル持続期間及び各サイクルの一部と関連するコールバック機能を使用することにより呼び出される、請求項40に記載の分配機。、

【請求項42】 前記選択は得ることができるプロセッサ時間量を基に行われ、選択されたレベルは得ることができるプロセッサ時間量内に調和し得る各サイクルの最大位置と関連するレベルである、請求項40に記載の分配機。

【請求項43】 コンピュータ・システムにおいて少なくとも2つのタスク間でプロセッサの時間配分を調整するための装置であって、

(a) 前記プロセッサの時間を一連の第1の期間に分割し、第1のタスクの前記プロセッサでの実行を開始し、各第1の期間の一部を前記第1のタスクに配分する手段と、

(b) 前記プロセッサ時間を前記一連の第1の期間と同じであるか又は異なる一連の第2の期間に分割し、高品質のパフォーマンスを提供するレベルであって各第2の期間を高レベルの使用で実行する第2のタスクの前記プロセッサでの実行を開始し、各第2の期間の一部を第2のタスクに配分する手段であって、前記第2のタスクは前記高レベルに比べ低いプロセッサ時間を使用する低いレベルであって前記高レベルに比べ低品質のパフォーマンスを提供する低レベルの使用でも各第2の期間を実行することができる手段と、

(c) 第1のタスクが配分されたプロセッサ時間に比べ期間において大きいプロセッサ時間を供給しているとき、第2のタスクを低レベル使用に切り換え、第1のタスクで利用可能にすべくプロセッサの時間の少なくとも一部を再配分する工程とを含む、調整装置。

【請求項44】 第2のタスクの低レベル仕様への切り換えは、一連の第2の期間のそのときの各期間の終わりに発生し、引き続く期間のために効果を与える、請求項43に記載の装置。

【請求項45】 第1のタスクが配分時間よりも期間において大きいプロセッサ時間を供給しているか否かの決定はタスクが前記期間に要求している時間の

推定を基にした予測により行われ、切り換え及び再配分は第1のタスクが前記期間の配分時間を消費する前に行われる、請求項43に記載の装置。

【請求項46】 第2のタスクの低レベル使用への切り換えは、前記一連の第2の期間のそのときの期間の終わりに発生し、引き続く期間のために効果をもたらす、請求項45に記載の装置。

【請求項47】 さらに、(d) 高品質のパフォーマンスを提供するレベルを有しかつ高レベルのプロセッサ使用時間を実行する第3のタスクのプロセッサでの実行を開始する手段であって、第3のタスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる手段と、

(e) 第1のタスクが配分されたプロセッサ時間より大きい時間を要求している場合に、第2のタスクを第3のタスクよりもむしろ低い使用レベルに切り換える先に作られた選択を基に、第3のタスクよりもむしろ第2のタスクを低使用レベルに切り換える手段とを含む、請求項43に記載の装置。

【請求項48】 コンピュータ・システムにおいてプロセッサの使用時間をタスクにより調整する装置であって、

(a) プロセッサの時間を一連の期間に分割する手段と、

(b) 高品質のパフォーマンスを提供する高いレベルのプロセッサ使用時間で実行するタスクのプロセッサでの実行を開始する手段であって、前記タスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低い品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる手段と、

(c) 前記各期間の一部をタスクに配分する手段と、

(d) 前記タスクが配分されたプロセッサ時間より多いプロセッサ時間を要求しているとき、そのタスクを低使用レベルに切り換える手段とを含む、調整装置。

【請求項49】 前記タスクは次の期間のために低使用レベルに切り替えられる、請求項48に記載の装置。

【請求項50】 前記タスクが配分されたプロセッサ時間より期間において

大きいプロセッサ時間を要求していることの決定は前記タスクが前記期間において要求している時間の推定に基づく予測により行われ、前記切り換えは前記タスクが前記期間において配分された時間を消費する前に行われる、請求項48に記載の装置。

【請求項51】 コンピュータ・システムにおいてプロセッサの使用時間をタスクにより調整する装置であって、

(a) プロセッサの時間を一連の期間に分割する手段と、

(b) 高品質のパフォーマンスを提供する高レベルのプロセッサ使用時間で実行する第1のタスクのプロセッサでの実行を開始する手段であって、前記タスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる手段と、

(c) 前記期間の一部を第1のタスクに配分する手段と、

(d) 高品質のパフォーマンスを提供する高いレベルのプロセッサ使用時間で実行する第2のタスクのプロセッサでの実行を開始する手段であって、前記第2のタスクは前記高レベルより低いプロセッサ時間を使用するレベルであって前記高レベルより低品質のパフォーマンスを提供する低レベルのプロセッサ使用時間でも実行することができる手段と、

(e) 第1のタスクが配分されたプロセッサ時間より期間において大きいプロセッサ時間を要求しているとき、第1のタスクが配分されたプロセッサ時間より大きいプロセッサ時間を要求している場合に、第1のタスクよりもむしろ第2のタスクを低レベルの使用に切り換える先に作られた選択を基に、第1のタスクを低レベル使用に切り替えるよりもむしろ第2のタスクを低レベル使用に切り替える手段とを含む、調整装置。



**【発明の詳細な説明】****【0001】****【技術分野】**

本発明は、1つのデジタル・プロセッサに利用可能な時間を2以上の選択的タスク又はスレッド分配し、それらの実行のスケジュールを作成する装置と方法に関する。

**【0002】****【背景技術】**

ソフト・リアルタイム要件を備えるプロセッサのスケジュール作成の必要性は、スマート(SMART)、リアルト(Rialto)、及びプロセッサ・キャパシティ・リザーブズ(プロセス容量予約装置)を含むいくつかの先行技術システムによって取り扱われてきた。

**【0003】**

スタンフォード大学設計のSMARTスケジューラは、従来及びソフト・リアルタイムのアプリケーションの同時実行を支援する。対話式アプリケーションは、よい応答を得る。それは、改良されたベスト・エフォート・スケジューラを使用し、軽負荷ではすべての最終期限が満たされ、過負荷では、低順位のタスクへのサービスを否定することにより、ゆるやかにパフォーマンスを下げるようにタスクの優先順位が使用される。それは、実行するタスクに向かって前進される規約時間の観念を利用している。

**【0004】**

マイクロソフト・リサーチで設計されたリアルト・スケジューラは、割り振られた環境で従来及びソフト・リアルタイムのアプリケーションの双方の同時実行を支援する。リアルトのアプリケーションは、独立して発案されているので、タスクがそのスケジュール作成の必要性を判断できるように、タイミングとスケジュール作成のモジュール性が必要とされる。リアルトは、割り振られたアプリケーションにとってEarliest Deadline First (EDF) (最早期限最優先)よりも優れた最小弛緩スケジューラを使用しており、中間パイプラインでアクセスされた構成部分のための最終期限の概念はない。リアルトは、始期拘束/終期拘束機

構と、利用、スケジュール作成を制限し、過負荷状態を処理するための予約装置の組み合わせを使用している。それらは「大抵の負荷除去状況を積極的に避ける」。

#### 【0005】

CMU (Processor Capacity Reserves) で開発されたプロセッサ・キャパシティ・リザーブ装置もまた、従来及びリアルタイム作業の双方の同時実行を支援する。アプリケーションは、必要な時間を前もって予約し、装置は、タスクが予約を超えないこと、また予約が装置の容量を超えないことを確実にする。予約は各期間に使用される時間量に関して与えられ、スケジュール作成はEDF（最早期限最優先）である。予約は、処理境界から境界まで通されることでタスクの割り振りを支援する。プロセッサ・キャパシティ・リザーブでは、スケジュール作成とQOSマネージャーが明確に分かれている。

#### 【0006】

##### 【発明の概要】

本発明に係るプロセッサ資源分配装置は、1つのプロセッサで多数のタスクに時間を配分する。それは3つの構成要素から成る。第1の構成要素は資源管理機（資源マネージャ）と呼ばれ、どのタスクにどれだけの時間量が与えられるかを決定する。第2の構成要素はスケジューラと呼ばれ、各タスクが配分された時間を得るように、タスクに時間を与える。第3の構成要素はポリシー・ボックスと呼ばれ、全てのタスクの要求を満たす時間が不十分な場合に、ユーザーと資源マネージャとに連絡する。

#### 【0007】

プロセッサは、1組の装置に見えるタスク・セット等のタスクのセットをホスト機かあるいは直接ユーザーに対して稼働するよう求められる。ホスト機はDOSボックスでもよく、コンピュータ稼働ウィンドウズか、他のものでもよい。あるいは、プロセッサは、ホスト機が入っていないセットの最上ボックス内に入れられていてもよい。APCユーザーは、オーディオ・ビデオ、モデム及びグラフィック装置のための支援を提供されてもよい。

#### 【0008】

何らかの装置に支持された1組のタスクは、(DOSボックスの場合のように)静的であってもよいし、(ウィンドウズ、又はジャワのアプレットのある場合はセットの最上ボックスの場合のように)動的であってもよい。これらのタスクは、同時稼働の装置の外観を呈するように、リアルタイムにプロセッサを共有しなければならない。

#### 【0009】

プロセッサ資源分配装置は、タスクを許容しスケジュールする役割を有する。それは、タスクが稼働できるか否か、いつ稼働できるか、またどの位の利用可能な時間を消費するか(各セッションにおいてどの位の稼働時間が可能か)を決定する。それは、1組の実際の物理装置が支援されていることをユーザーが信じ続けられるように、プロセス時間の使用を管理する。多数の装置がシミュレートされている事実は目に見えない。

#### 【0010】

この環境では、ソフト、リアルタイム要件を有する。タスクには最終期限があるが、最終期限に間に合わない場合、結果は不運ではあるものの、破滅的ではない。この限られた環境では、従来のような作業場又は時分割タスクを持っていないかもしれない。もっとも、セットの最上環境ではある程度それらがあるかもしれないが、非周期的タスクは通常、頻繁に実行されないコマンドに関連する。

#### 【0011】

##### 従来技術装置との対比

許容制御が保証されている。本発明の装置では、タスクが許容を許可されると、それが終了されるまで少なくともそれ自体の規定最小回数を得るまで保証される。

#### 【0012】

資源の配分は、ステップ関数である。他の装置は資源配分を平滑関数として行う。他の装置においては、優先順位は過負荷の場合稼働タスクのパフォーマンスを「ゆるやかに下げる」ために使用される。この取り組みの問題点は、本発明のタスクはいずれも、その利用がわずかに変更される場合は「ゆるやかに下がらない」ことである。むしろ、資源配分における対応する段階を必要とするサービス

の質には、はっきりと段がある。配分がこれらの段の1つに行われない場合は、より低いレベルの質の方に資源を浪費するか、あるいはより高いレベルのための最終期限を逸するかのどちらかの結果となる。

#### 【0013】

許容制御と許可セットのための最小再計算。本発明の実施例においては、スケジュールに悪影響を及ぼすような事象が発生したときに、またそのときにのみ、スケジュール作成情報を再計算する。始期拘束／終期拘束対を使用するリアルトのような装置は、各周期に関する各タスクに対するスケジュール作成の可能性を再計算する。

#### 【0014】

本発明の実施例においては、装置が過負荷の状態の場合、又は最終期限が見逃されそうな場合、スケジュール作成を再計算しないし、アドミタンスについての取り組みを決定しようとししない。これはリアルトのような装置の取り組みである。それは、スケジュール作成の決定が行われつつある時期までには、最終期限が既に見逃されているかも知れないという欠点を有する。本発明の実施例においては、最終期限は決して見逃されない。

#### 【0015】

本発明の実施例においては、休止タスクを支援する。これらのタスクは、容認制御を再計算するコストをかけず、休止状態をやめるまで資源を消費しないで、稼働したいときに容認を保証される。ある装置は、容量の予約のためにこの種のタスクを、使用されていないものに至るまで全量を必要とするものもあるであろう。その結果、リソースが浪費される。他の装置は、実際に実行を望むときにタスクが判断することを要求するものもあるが、その結果、既に容認されている他のタスクが終了されてしまう。

#### 【0016】

どのタスクが負荷を除去するか、あるいは最終期限を逸するかに関する全体的な取り組みの決定。他の装置は、過渡的な過負荷が検知されたときに全体的な決定をするが、その時点で何らかのタスクがすでに最終期限を逸している。事実上、負荷を除去することが求められてきた。そして、過負荷が発生したら次にどの

タスクが偶然稼働するか、このタスクの「選択プロセス」はほとんどでたらめである。

#### 【0017】

本発明の一実施例は、プロセッサで行われるタスクの容認を制限する方法である。各タスクは、高品質の実行を与える高レベルと低品質の実行を与える低レベルとを含む2以上の時間使用のレベルを有する。タスクの最低使用レベルの合計がプロセッサで利用可能な全時間を超えない限り、タスクは処理のため容認される。1つのタスクが処理を必要とするがプロセッサで時間が利用不能であれば（既に容認されたタスクの最低使用レベルの合計が大き過ぎて新しいタスクを容認できない場合）、新しいタスクは除外される。しかし、さらに低い最低使用レベルなら、その使用レベルが利用可能時間内に適合できる範囲であれば容認されるであろう。

#### 【0018】

タスクが一旦容認されると、実行が開始される。タスクの1つが一時的に実行を必要としなくなり、それによってプロセッサにおいて利用可能な時間が増える場合、最高使用レベルより低いレベルで実行されてきた他のタスクは、解放された時間を使用するため、より高い使用レベルにその実行を切り替えるであろう。また、あるタスクが一時的に実行を必要としなくなり、プロセッサで未使用の時間が利用可能になった場合、本発明の方法の重要な面は、一時的に実行を必要としなくなったそのタスクも含めてすべてのタスクの最低使用レベルの合計が新しく容認されたタスクに時間を許さない限り、追加的なタスクにその利用可能時間を許さないことである。すなわち、1以上のタスクが実行を必要としなくなった場合に十分な時間がプロセッサで利用可能であるという事実だけでは、未使用の時間の使用を別のタスクに許すには不十分である。その代わり、残りの容認されたタスクの最低使用レベルの合計が、プロセッサで利用可能な時間を超えることなく新しいタスクが容認されるに充分小さくなるように、先に容認されたタスクの1つが終了されなくてはならない。

#### 【0019】

本発明の別の面において、本発明の方法は、アプリケーション・プログラム作

成インターフェイスを複数のアプリケーションに提供する。各アプリケーションは、プロセッサ資源分配装置に対してそのアプリケーションを呼び出す2以上の機能を指定する。この方法においては、プロセッサ資源分配装置は、プロセッサで利用可能な時間に基づくタスクの呼び出しに使用される機能を選択して、各タスクをその機能の1つと共に呼び出す。アプリケーションがその呼び出し機能をプロセッサに提供する際に、各アプリケーションは呼び戻し機能のリストと、各呼び戻し機能に関連して、リアルタイムの分割のための周期持続性（サイクル持続期間）、及び各周期持続性に対して、タスクがその機能と共に呼び出された場合、そのタスクによって使用される各周期（サイクル）の一部の仕様を提示する。その情報は、周期持続性と、使用される各周期の部分とから成る。この情報は、処理時間を各アプリケーションに配分するスケジュールを決定するために、プロセッサ資源分配装置によって使用される。

#### 【0020】

本発明の別の面は、1つのタスクから別のタスクへと時間を移行させる方法を含む。時間使用の少なくとも2つのレベルを有する第1のタスクが、仕様のより高度なレベルで操作しており、特定の期間のために第2のタスクがタスクに割り振られてきた時間よりも多くの時間を要求するならば、本発明は第1のタスクを、第2のタスクによって時間を利用可能にする、より低い使用レベルに移行させる方法を含む。さらに、1つのタスクから別のタスクへ利用可能な時間を移行させる代わりに、第2のタスクは、特定の期間又は周期に処理を終了するには十分な時間がなかったことをプロセッサに報告することができる。そうすれば、他のタスクが先に終了するときに利用可能になる時間を与えられるであろう。もし次の周期又は期間の開始前に完了するには不十分な時間が利用可能になる場合、この事実がリソース・マネジャーに報告されれば、リソース・マネジャーは、より低い使用レベルを指定する呼び戻し機能を有する次の周期のタスクを呼び出すことができる。

#### 【0021】

##### 【発明を実施するための最良の形態】

第1原則：図1に示すように、プロセッサ資源分配装置1は、以下の3つの第

1 原則に従う。

【0022】

1. 一旦タスク2がユーザーによって首尾よく開始されると、ユーザーの見通しから、そのタスクは自然に終了するまで、あるいはユーザーがそれを終了するまで継続する。

【0023】

ユーザーは、何らかのコマンド回線、対話、ボタン又はその他何でもを介して（例えば、CD再生装置の演奏ボタンをたたいて）、タスクを開始する。一旦タスクが開始されると（例えば、サウンド・トラックがスピーカーから出始めると）、それが終了する（CDの最後に来る）まで、あるいはユーザーが（CD再生装置の「停止」ボタンを押して）それを終了させるまで継続する。プロセッサ3のこの機能性に関わるタスクは、CDの演奏中連続して稼働しないかも知れない（実際、理想的には、それらタスクは連続して稼働しないであろう）が、ユーザーはこれをスピーカーの音からは検知することはできないであろう。

【0024】

この原則は、過渡的過負荷があるときに稼働中のタスクを終了させる可能性のある装置では満たされない。

【0025】

これは、タスクが強制排除なしに稼働することを保証するのではなく、連続的な機能性の出現を維持するに充分頻繁に充分な時間を許可されることを保証する。プロセッサ分散装置1は、タスク2が時間使用の態勢にある限り、プロセッサ3に利用可能な時間を（ほとんど）100%配分する。

【0026】

2. タスク2が稼働態勢にあり、再現時間周期（図2における「期間」）の幾分か未使用であれば、その時間量が稼働態勢タスクに利用可能にされる。すなわち、プロセッサが部分的にアイドルであり、タスクがさらにプロセッサ時間を要求するならば、それはさらに時間を許可される。これは、1つのタスクがそれ自体に超過プロセッサ時間を蓄える必要がないことを意味する。先のタスクが蓄えておいたが使用していないいくつかの時間を別のタスクが要求するなら、後の

タスクはその時間を与えられる。

#### 【0027】

3. サービスの質の修正は、ユーザーの要求事項を参照して行われ、そのような手段決定は、全装置の了解でなされる。

#### 【0028】

プロセッサの時間があまりに多くタスク・セットによって要求されるために装置が過負荷になると、あるタスクは終了されるか、より低い質のサービスを提供することによって負荷を下げるように要求されねばならない。どのタスクに負荷を下げるように要求すべきか、またそのサービスをどの程度下げるよう要求すべきかの決定は、ユーザーの選択に基づいてなされるべきである。タスク・セットが如何にユーザーの必要事項を満たすかについて、包括的な（装置全体の）了解がなければならない。また、サービスの質の決定は、単独のタスクの必要事項を理解するだけで行われてはならない。

#### 【0029】

##### プロセッサ資源分配装置の設計原則

プロセッサ資源分配装置1の設計にとって重要な設計選択事項がいくつかある。

#### 【0030】

##### スケジュール作成と許容制御の分離

この実施例においては、どのタスクを稼働するか（資源管理装置5）、及びどのタスクを現在稼働中であるか（スケジューラ4）の決定を別々に行う。

#### 【0031】

どのタスクを稼働するか（またそれらを如何によく稼働するか）の決定プロセスは、資源管理装置5によって行われるが、さらに2つの構成要素に分けられる。第1の許容制御6は、タスクが装置に許容され得るか否かを我々が決定するためのプロセスである。許容されたタスクは、プロセッサ時間の最小量を保証される。アプリケーション書き出しプログラムは、ゼロを最小と指定するオプションを有する。第2の許可制御7は、各タスクにどの程度のプロセッサ時間を与えられるかを決定する。その許可は、消費できる期間とその量とからなる。例えば、



1つの許可は、30ms 期間に10ms を配分することができる。許可は、これだけの時間が各指定期間にタスク2に配分されるという、タスク2への保証である。

#### 【0032】

実際にタスクに稼働せよと命令するプロセス、及び許可された時間の配分が引き渡されたと保証するプロセスは、スケジューラ4によって行われる。スケジューラは手段決定をしない。それは、許可制御プロセス7によって設定されたタスクのセットを介して単に循環するのみである。

#### 【0033】

##### ユーザー制御のスケジュール作成手段

当社は、どのタスクが実行されるべきかを決定する手段と、その選択を行うのに使用する手段とを分離している。同様に、当社は、特定のタスクがどの程度の質のサービスを提供するべきかを決定する手段と、その質のサービス・レベルを選択するのに使用する手段とを分離している。

#### 【0034】

ユーザー（又はホスト機）8は、スケジュール作成にどのタスクが選択されるか、及びどのレベルの質のサービスを各タスクが提供を要求されるを制御する。資源管理装置5は、スケジュール作成手段の決定が行われなければならない場合に、ユーザーに操作されたスケジュール作成手段情報をポリシー・ボックス9にアクセスするためのフックを有する。

#### 【0035】

プロセッサ資源分配装置1は、どのタスクのセットが稼働するか、又はどの程度に稼働するかを決定することに運と時間調節が関与しないように設計されている。配分手段の決定は、常に明白になされる。これは周期タスクの寿命における各期間に対する新スケジュール作成決定を効果的に行う装置と対照をなす。当社は、新しいタスクがスタートしたか停止した、あるいはタスクがその利用パターンを修正する必要があるといった、資源割振に重要な事象が発生したときに、手段配分決定を行うだけである。

#### 【0036】

プロセッサ資源分配装置1の機能は、3つの主要構成要素であるスケジューラ4、資源管理装置5、ポリシー・ボックス9に分かれる。ポリシー・ボックスは、装置設計者によって予め設定されているが、ユーザー8が個人的な選択事項に合うように修正してもよい。例えば、デフォルトにより、ビデオ処理は、オーディオ処理より質が低下するが、聴覚障害ユーザーは、その逆にしたいかも知れない。

#### 【0037】

##### プロセッサでの時間の割り振りの概要

資源分配機1の機能は、スケジューラ4、資源マネージャー5及びポリシー・ボックス9の3つの主要素に分割される。

#### 【0038】

資源マネージャー5は、容認制御6、許可設定7、及びユーザーのために多重タスク間における利用時間協定のために応答することができる。

#### 【0039】

スケジューラ4は、資源マネージャー5により作られた容認を受け取るスケジュールを実行し、時間の配分を実行し、実際の利用を資源マネージャー5に知らせるために応答する。

#### 【0040】

タスク2がプロセッサ3を開始させることを必要としているとき、プロセッサの時間の要求が資源マネージャー5に対して行われる。プロセッサ・サイクル（プロセッサ周期）のための要望は、要求されたプロセッサ時間にわたる（繰り返し）期間とプロセッサ時間が各期間において必要とする時間量とにおいて与えられる。例えば、MPEGデコーダ・タスクは、各30ミリ秒のプロセッサ時間のうち10ミリ秒を必要としてもよい（図2参照）。

#### 【0041】

もし要求された時間のために十分な超過容量であると、資源マネージャー5はタスク2により作られた要望を容認する。そのタスクは、スケジュールを作成されたセットに加えられ、タスクがユーザー又は他の外部事象により終了されるまでそれらをとどめる。

## 【0042】

もし過負荷状態が発生すると、ある特別なアクションがなされなければならない。この場合、資源マネージャー5は、タスクがある負荷の切り離しを要求されねばならないことを決定するようにポリシー・ボックス9を呼び出し、低品質のサービスを提供する。これには、連絡することができないという要求を作られたタスクが負荷を排除する（捨てる）ように質問される1つである、という含みはない。むしろタスクの独立した秩序転移及びそれらのサービスの質は、この決定を援助するようにポリシー・ボックスにより提供される。

## 【0043】

本実施例においては、厳格な容認制御を行うという事実の含みは、スケジューラ4がスケジュール可能のタスクセットだけのスケジュールを作成するように問い合わせられることを保証するということである。これは、Earliest Deadline First (EDF)のようなサンプル・スケジューリング・ポリシーを使用することができる、ということの意味する。EDFは、確実な制約で、理論的にスケジュールを作ることができる適宜なタスクセットのスケジュールを効果的に作成するために提供される。そのような制約は、優先使用の可能性と、タスク間での同調の欠落とを含む。より小期間のタスクは、長い稼動タスクのために単一期間の間、ある程度の時間で稼動される。これは、長いタスクが優先使用可能であることを必要とする。スケジューリングは、タスクが期間のある時点でそれらの使用を許すことができるならば、保証され得ない。これは、周期的なタスクがそれらの期間内で同期することができない、ということの意味する。プロセッサ資源分配機、制御ビデオ、オーディオ、及び通信のための典型的なタスクセットの周期的タスクは、それらの要求に遭遇する。

## 【0044】

この作業の手法

タスクの稼動を終了させない目的のために、その時間を使用するか否かのタスクのために十分な時間を予約しなければならない。システムが過負荷になっていると、タスクがその利用時間を低減することを要求する手法を必要とするが、要求を否定してもよい。

## 【0045】

これら潜在的な矛盾する目標に遭遇するために、表1に示すような資源リストを導入することができる。このデータ構成は、支援可能な各サービス・レベルでタスクのために要求される時間利用の明細すなわち仕様を示す。各エントリーは、期間、利用性、率（レート）及びファンクション（機能）を含む。期間は、ミリ秒（m秒）又はシステムの時間単位において、タスクが遂行しなければならないサイクル期間の明細を示す。利用性は、各期間において消費される時間量（ミリ秒又はシステム時間単位）の明細を示す。レートは、期間により分割された利用性として計算され、またこのタスクにより使用される時間の百分率の値である。機能パラメータは、特定の期間／利用性の結合がタスクに容認されているとき呼び出される機能の明細を示す。タスクは、期間及び利用性が呼び出される機能を基に容認されるものを決定する。

## 【0046】

タスクが単一の低サービスの支援だけを行うことができるならば、それはその資源リストにおける1つのエントリーだけを有する。その表内にいくつかのエントリーがあるならば、それらの表はレートにより規則正しくされる。最大レートのエントリーはタスクのための最良のパフォーマンス(performance)を表し、最小レートのエントリーはタスクが最小レートにより合理的に得ることができる最小処理量を表す。アプリケーションは、資源マネージャー5を呼び出すことによって資源リストのエントリーを変更することができる。

## 【0047】

タスクに容認された実際のレートは、テーブル内のエントリーの1つである。資源マネージャーは、結合がその資源リスト内に存在している限りタスクに対し期間／利用性の結合を許容しない。

## 【0048】

## 【表1】

期間 (m秒)	利用性 (m秒)	レート (計算済)	ファンクション	
期間 <sub>m</sub>	利用性 <sub>m</sub>	利用性／期間	ファンクション <sub>m</sub>	← 最良
期間 <sub>1</sub>	利用性 <sub>1</sub>			
期間 <sub>k</sub>	利用性 <sub>k</sub>			← 許可
期間 <sub>j</sub>	利用性 <sub>j</sub>			
期間 <sub>i</sub>	利用性 <sub>i</sub>			← 最小
.....	.....			
期間 <sub>a</sub>	利用性 <sub>a</sub>	利用性／期間		

表1：資源リストのフォーマット

## 【0049】

各タスクは、少なくとも1つのエントリーを持つ資源リストを資源マネージャー5に提供する。アプリケーションは、スケジューラ4がこのタスクを遂行しているとき、実行すべきコールバック機能を提供する。キャリブレーション期間の間、利用性はユーザーにより省略することができ、資源マネージャー5はそれを充填する。

## 【0050】

資源マネージャーの実行内容

資源マネージャー5は、容認制御及び確立された容認セットのために応答可能である。

## 【0051】

資源マネージャー5は、各タスクに与えられる時間配分を決定する。各タスクに与えられる時間配分は、「容認」を呼び出される。容認は、タスクの資源リストにおけるエントリーの1つに応答される。

## 【0052】

タスク用の容認は、3つのイベントの1つが発生するとき、変更することができる。最初は、タスクがシステムに入るとき又はシステムから離れるときである。第2は、タスクがその資源リストを変更するときである。第3は、あるタスクが「静止」状態に入るとき又は「静止」状態にあるときである。静止状態は、今

後稼動を希望し、それ自身のためにプロセッサ時間の配分を希望するタスクのために予約され、その結果容認層(admission later)を否定しないことがある。しかし、これらのタスクは、それが静止しているとき、その間多くの時間を使用しないから、他のタスクへの使用のためにそれらの時間配分が自由になる。

#### 【0053】

図3に示すように、新たなタスクが立ち上がると(ステップ21)、そのタスクはその資源リストで資源マネージャーに接触する(ステップ22)。資源マネージャーは、次に容認制御を行う。新たなタスクは、システムにおける全てのタスク(現在実行可能のものと静止しているものとの両者)の最小容認の和が100%以下であるときだけシステムに入ることを許される(ステップ23)。

#### 【0054】

タスクが、システムに入るか(ステップ22)又はシステムから離れるとき、若しくは、潜在的に静止しているタスクが状態を変えるとき(ステップ25)、資源マネージャーは全ての容認されたタスクの許可の新たなセットを発生しなければならない。この時点で、タスク用の許可を増減させてもよい。このプロセスの最初のステップとして、全てのタスクがそれらの最大許可を得ることができるか否かを決定する(ステップ26)。それができないと、ポリシー・ボックスがユーザーの性能優先(performance preference)を得るようにアクセスされ(ステップ27)、妥協した許可時間が計算される(ステップ27)。達成された新たな許可セットで、次のステップは資源量を自由にすべくスケジューラで現在の許可を除く又は低減させ(ステップ29)、最終ステップは新たな及び増大された許可をスケジューラに通知する(ステップ30)。タスクは、次の期間の利用が新たな許可と関連する機能に対する呼び出しでもって開始されるので、間接的に通知される。許可の和が100%を越えないことを資源マネージャーが保証しているので、スケジューラ4は全てのタスクの引き続くスケジュールを作成すべく単純なEDF構成を使用することができるよう許可を実施するだけである。

#### 【0055】

##### 【数1】

容認制御:

$$\sum_{i=0}^{Runnable} rate(min)_i + \sum_{j=0}^{Quiescent} rate(min)_j \leq 100\%$$

【0056】

【数2】

許可セット：

$$\sum_{i=0}^{Runnable} rate(grant)_i \leq 100\%$$

【0057】

スケジューラの実行内容

スケジューラ4は、図4に示すように、どのタスクを次に稼働させるかを決定し、許可を実行し、資源マネージャーにアカウントをする資源を提供するために、応答することができる。

【0058】

スケジューラへの入力は、許可セットである。これは、許可された稼働可能なタスクの資源リストにおける入力セットである。スケジューラは、「次の期間が始期、次の期間の終期」のための一定の時間で「期間、利用性」のセットを増大させている。

【0059】

タスクの次の稼働を決定するために、スケジューラはEDFアルゴリズムを使用する。使用される最終期限は許可のための期間端すなわち終期である。この期間のための全許可を既に受けていない近くの次の終期を有するタスクは、いつでも選択される。

【0060】

次のコンテキスト(context)スイッチは次の3つの理由の1つのために発生する。第1に、タスクはプロセッサに（その許可を仕上げるまたは仕上げるこ

しに) ゆだねてもよい。第2に、タスクはその許可を越えてもよい。第3に、他の終期は現在の期間に先立って発生してもよい。

#### 【0061】

タスクが一度選択され、タスクに対するスイッチングの前に、スケジューラはタイマーの割り込みを設定することにより再度制御を得るために整える(ステップ48)。スケジューラは以下より早期にタイマー割り込みを設定する。

#### 【0062】

選択されたタスクのための許可終期、又は、まもなく到来する次の期間終了時を有するタスク(現在のタスク以外の)用の次の期間開始時。

#### 【0063】

タイマーが第1のイベントにおいて行われると、タスクはその許可を超過される。タイマーが第2のイベントにおいて行われると、現在稼動している1つが新たな期間を開始するより短い期間を持つタスクは、新たな期間を開始し、稼動中のタスクよりはむしろ最終期限を有する。

#### 【0064】

許可の終わりに、又はタスクがプロセッサを譲渡し、この期間で実行していることを表しているとき、次の期間が開始され、次の終期が更新される。更新は、タスクが先取りしてもよいので、タスクがプロセッサに承諾されるときは行われない。

#### 【0065】

図4に示すように、プロセッサは、割り込みでもって開始し(ステップ41)、傑出した許可であるか否かのチェックをする(ステップ42)。優れた許可であると、時間がタスクに与えられる(ステップ47)、次の中断が設定され(ステップ48)、タスクが稼動される。優れた許可でないと、多くの資源量が最近要求されているならば、許可セットが更新される。これが優れた許可をもたらすならば、処理はステップ47に移行する。傑出した許可でないならば、システムは、その許可内で完了せずかつ超過時間を供給するタスクであるか否かのチェックをする(ステップ45)。そのようなタスクであると、タイマーはそのタスクに提供され(ステップ46)、ステップ48に続く。超過時間が要求されてい



いと、タイマーは散発的なサーバーに与えられる（ステップ50）。

【0066】

図2に示すように、以下は、モデム、3Dグラフィックス表示装置及びMPEG3デコーダの3つのタスク用のサンプル・スケジュールを示す。時間0では、複数のタスク用の許可セットは以下ようになる。

【0067】

【表2】

	期間 (m 秒)	利用性 (m 秒)	レート	ファンクション	始期	終期
モデム	10	1	10.0 %	Modem()	0	10
3-D グラフ ィックス	13.9	7	50.4 %	Frame_72Hz()	0	13.9
MPEG	30	10	33.0 %	FullDecomp()	0	30

表2：3つのタスク用の初期許可

モデム、MPEGデコンプレッション(decompression)、3Dグラフィック

【0068】

【表3】

				時間 5		時間 11		時間 29	
	期間	利用性	レート	開始	終了	開始	終了	開始	終了
モデム	10	1	10.0%	10	20	20	30	30	40
3-D グラフ ィックス	13.9	7	50.4%	0	13.9	13.9	27.8	27.8	41.7
MPEG	30	10	33.0%	0	30	0	30	30	60

表3：3つのタスク用の引き続く始期及び終期

モデム、MPEGデコンプレッション、3Dグラフィックス

【0069】

表3は、3つのタスクを稼動させるために、最初の70ミリ秒用のスケジュールを示す。66ミリ秒を超えるこのスケジュールは、図2に図表として示される。

。

## 【0070】

ここに、スケジューラの稼動のための疑似コードを示す。

## 【0071】

```
while (1) {
    task_to_run = find_task ( min "period_end" AND "period_begin" <=
time_now);
    if (preempting_task = find_task ("period_end" < task_to_run ("period
_end"))
        AND min ("period_begin"))){
        alarm_time = preempting_task ("period_begin");
    } else {
        alarm_time = time_now + task_to_run ("utilization");
    }
    set_timer (alarm_time);
    run (task_to_run);
    /* if the user returns here, they have finished their grant for this p
eriod*/
    task_to_run (period_begin)+= task_to_run (period);
    task_to_run (period_end)+= task_to_run (period);
}
```

## 【0072】

3Dグラフィックス・タスクは図示の5つの期間中の3つの期間に先取り権なしに稼動するが、MPEGデコーディング・タスクは少なくとも一回で常に先取する。

## 【0073】

#### ポリシー・ボックスの実行内容

ポリシー・ボックス9は、最高レベルのプロセッサ時間を与えるべき全てのタスクに与えるには不十分な資源であるとき、他に対して交換しなければならないタスク2に関する情報源である。

## 【0074】

支援されたタスク2のセット毎に、ポリシー・ボックス9は、ユーザーに対する各タスクの相対的な重要性を理解している。それらには、プロダクトで初期に移す格付けのデフォルト・セット(default set)がある。このセットは、システムのユーザー又はシステム管理者により無効にすることができる。

## 【0075】

それらには、実行され得るポリシー・ボックスのような各種の手法がある。実施例においては、セットの各タスクのためのプロセッサ時間の相対的配分に応答する百分率のセットを維持している。それら百分率は、資源リスト21に開示されている、タスクにより支援することができる現実のレートに関連しない。百分率セットは、システムが過負荷のとき、ポリシー・ボックス9により資源マネージャー5に戻される。

## 【0076】

容認された格タスクのために、資源マネージャー5は、ポリシー・ボックス9により与えられた百分率より高い又は低い2つの資源リスト・エントリー(Resource List Entry)を選択する。そのように選択された高エントリーの総和がプロセッサ時間の100%より小ささいと、全てのタスクはこの高エントリーを得ることができる。しかし、そのような和が大きすぎると、ポリシー・ボックス9によりそれに与えられた百分率とそのより高レベルとの間に大きな隔たりを有するタスクは、低レベルを与えられることになる。このプロセスは、保証の和が入手可能なプロセッサ時間の100%より小さくなるまで繰り返される。

## 【0077】

最終的な付加的なパス(pass)は、許可の和が100%未満であるときに、形成される。可能であるならば、100%の近くで使用されるように、タスクの許可が増大される。さらに、現在選択されたレベルに近い次の高いレベルを有するタスクは、最初に付加的なプロセッサ時間を与えられる。

## 【0078】

タスクの展望

ここで、スケジューラ4と資源マネージャー5がタスク2の外観から行うこと

を説明する。それには、正常タスクと静止タスクとの2種類のタスクがある。正常タスクは、稼動する準備ができているものである。静止タスクは、それらのスケジュールを正常に行うことを一時的に放棄されており、次の期間開始時を有していない。

#### 【0079】

##### 正常タスク

タスクは、これが最初に立ち上げられるとき、資源リスト21で資源マネージャーを呼び出す（ステップ22）。この時点で校正ループを稼動させてもよいし、レート用を除いて資源リストのエントリをファイルしてもよい。多くの場合、少なくとも機能が満たされていなければならない、やがては、期間、利用性及びレートが同様に満たされる。

#### 【0080】

タスクが制御を得る次の時間は図1に示すように、資源リストに与えられた機能の1つが呼び出されるときである。タスクは、3つの機能の1つを解することを除いて各種の手法で制御を得られない。その機能は、1フレームの計算、フレームセットの計算等、完了すべき作業の微少セットに相当する。フレームが慣性すると、その機能の復帰はプロセッサに効果的に与えられ、タスクはこの期間を計算する。

#### 【0081】

タスクは、同じ機能を通すか又は資源リストに与えられた他の機能を通して、そのタスクのために次の期間の時間を再度コールされる。各時間毎に、タスクは、3つの機能のうちの1つを通してコールされ、最後の期間における実際の利用性のレポートを与えられる。これは、資源リストを調整すべく選択的に使用される（期間を使用する持続時間はある2つの期間の差であってもよい。）。

#### 【0082】

タスクは、任意の時間でその資源リストを調整することができ、特定の最小レートが先の最小レートより大きくない単一のケビエート(caveat)で、システムに容認を許可される。エントリは加えられ、除かれ、又は変更され、さらには新たな機能を自由に特定される。

## 【0083】

タスクがその許可を越えると（例えば、許可が超過する前に譲渡されないならば）、タスクは直ちに先取りされ、そのタスクはそれが稼動する次の時間に超過して稼動することを通知される。次の稼動期間において、タスクが稼動する次の期間において、そのタスクは、同じ処理を継続すること、負荷を切り離すこと、又は適切なエルスを選ぶことができる。図4のステップ45及び46に示すように、資源が過剰であると、その許可が期限切れになる前に計算されない多くのタスクは、稼動することを許可される。そのようなタスクが2以上であると、最初に早期に期限切れになるタスクがその最終期限に達するまで稼動することを許される。しかし、アプリケーション・ライターは、許可を越える特別な資源を立案しない。

## 【0084】

静止タスク

システムは、直ちに稼動する必要のないタスクであって稼動する必要があるとき容認を否定されないタスクのための時間を予約することができる。そのようなタスクが資源の要求を資源マネージャー5に登録されるまで待つと、タスクセット用の最小時間の和は既に100%になる（ステップ23）。次いで、そのタスクを否定するか（ステップ20）、既に稼働中のタスク（第1の原理の1つに違反するタスク）を切り離す。これの代わりに、静止させることを実際に希望しているときにタスクを適宜なアイドル・ループに回してもよい。しかし、これは他の第1の原理に違反し、それらを使用することができるタスクがあると、機械時間の100%を使用することになる。

## 【0085】

この課題は、静止タスクの概念を導入することにより解決することができる。開始時、このタスクは、通常のように資源マネージャーに登録する（ステップ22）。その資源リスト21は、それが実際に稼動しているとき必要な時間に匹敵し、静止状態のために特別にはエントリーしない。その最小必要条件は、容認のために使用してもよい。資源マネージャーへの呼び出しのアーギュメント(argument)は静止タスクを確認し、そのタスクはシステムに一回容認されると一時停止

される。

【0086】

休止は、静止と本質的に異なる。休眠（ブロック）が周期的タスクとして実際にスケジュールをされないタスク、以下に記載するスポラディック・サーバ(Sporadic Server)は、休眠タスクのスケジュールをすることができるが、プロセッサ資源分配機はブロッキングをされている多くの期間においてタスク用の保証スケジュールを作成することができない。静止タスクは、資源マネージャーを介して配分される1つであるが、静止の間スケジュールをされない。

【0087】

タスクが外部事象により静止から目覚めさせられると、保証は再計算されるが、容認制御作業は既に行われている。静止タスクが立ち上がると、他のタスクは、次の期間に新たに目覚めさせられたタスクを稼働させるべく不十分な時間であるより小さい許可を与えられる（ステップ29）。しかし、容認制御式のために利用可能な時間にふさわしい許可セットが保証される。

【0088】

許可セットを再構成しかつタスクを浮遊させる資源マネージャーに対する呼び出しにより（あるタスクが稼働できるように）静止タスクを静止状態に戻してもよい。

【0089】

正常タスクの例：MPEG2ビデオ・デコーディング

MPEG2ビデオ・デコーディングは、正常タスクの一例である。これは、MPEG標準により規定されたフレーム・リフレッシュ・レートである30ミリ秒の期間を有する。本実施例で使用するプロセッサにおいては、プロセッサ時間のほぼ33%がこのタスクに使用される。

【0090】

MPEGにおけるコーディングには、非常に異なる利用性を有する改良された3種のフレームが用いられる。最初のフレーム(I)は、最小の圧縮を有しており、最小のプロセッサ時間を必要とする。Pフレームは、Iフレームから計算され、中間のフレーム時間を必要とする。Bフレームは、前の一对のI又はPフレー

【0 0 9 1】

【0092】

	期間 (m秒)	利用性 (m秒)	レート	機能
最良	30	10	$10/30 = 33.3\%$	FullDecompress()
	120	30	$30/120 = 25\%$	Drop_B_in_4()
	90	20	$20/90 = 22.2\%$	Drop_B_in_3()
最小	120	20	$20/120 = 16.7\%$	Drop_2B()

フレームの負荷切り離し：  $1/4$ ，  $1/3$  又は  $1/5$

【0093】

【0094】

この資源リストは全フレームが同じ方法で圧縮解除されるMPEGタスクの要求を含み、単一フレーム内の利用性を変更する必要はない。より短い期間を有する他のタスクがあるならば、そのタスクがある機能で呼び出されるとき、最も短い期間でさえ先取りされるように確かめられる。それらは異種の直交法であり、

最適な先行取得が発生するとき調整するように使用することができる。このことは後により詳細に言及される。

#### 【0095】

MPEGタスクがある理由のめにその許可を超過すると、それは許可され、次の期間の通知を受ける。フレームの種類に応じて制御を回復しているとき行われるアクションは、圧縮されている。そのタスクがBフレームであると、それを下げるように選択され、次のフレームに継続する。タスクがIフレームであると、それを終了し、次のBフレームを下げるように特筆されるであろう。正しいものはどれでも、MPEGタスクはその決定をする。タスクが一貫してその許可を超過していると、その資源リスト内の機能と関連する利用性の要求を変更することが必要になる。

#### 【0096】

##### 休止タスクの例：電話への応答

ユーザーは、電話に応答し電話をかけた人とのモデム接続を確立するタスクを有することができる。電話が鳴るまで、そのタスクにはすることがない。電話が鳴ると、タスクは接続をしてそれを維持するに十分なプロセッサ時間を配分されなければならない。

#### 【0097】

配分の見込みがあるので、このタスクは稼働したいときに容認制御が容認を拒否することができないタスクである。実際、そのためには時間を取っておく必要がある。休止タスクでのみこれを行うことができる。

#### 【0098】

電話が鳴る前のいつでも、電話応答タスクは資源マネージャーに、稼働中の最小要件を確立して登録する。電話がかかって操作系がタスクを起動させるまでそれは中断している。資源マネージャーは、許可セットを再計算するが、容認制御アルゴリズムのため運転可能なスケジュールを見つけることを保証されている。電話応答タスクが稼働していない場合、その時間は、装置内の他のタスクにより高いレベルのサービスを提供するために利用率することができる。

#### 【0099】



処理超過時間：負荷移行

許可配分方法及びスケジュール作成は、期間毎に必要な時間が比較的一定である場合に行われるのが理想的である。MPEGやオーディオ等いくつかの周期的なタスクは、このモデルに非常によく適合する。処理要件が、画面上の画素数のように、決まった数値によらず、1シーンにおけるポリゴンの数のように非常に変化しやすい数値による、3Dのような他のタスクもある。大抵の3Dタスクは、この課題を制限しようとするポリゴン・バジェットを有するが、フレーム毎に5,000から10,000ポリゴンを有する現行のバジェットは、いまだに2:1の処理能力の偏差を残している。

## 【0100】

3Dアプリケーション設計者が10,000ポリゴンを処理するに十分な配分を要求しても、この数を超えるフレームがまだあり得る。また、もちろん、フレームの大部分は、5000ポリゴンしか使用しないならば、3Dに残しておいた処理容量の約半分が浪費される。

## 【0101】

この課題にアドレスするために、負荷移行の観念が発明された。多数の装置は、タスクが負荷を下げることで、多分、1フレームのための計算の途中で部分的に処理されたポリゴンを下げることを予期している。そのような脱負荷に加えて、限られた持続時間に1つのタスクから別のタスクへ時間の変換を制御されている負荷移行をも許されている。このため、別のタスクの処理能力を損なわずに、必要に応じて余分の時間を利用率できるようにする一方、正規の条件で3Dのようなタスクに時間量をより少なく配分することが可能になる。

## 【0102】

負荷移行の可能性は、先ずポリシー・ボックス9に指示される。タスクのセットがつながれ、1つのタスクがその許可を超えるかも知れないことを表示する。超過の事象においては（既に過ぎたか予期されるか、いずれか一方）、スケジュール4はこの1つの期間にセットのうちの他のタスクから超過したタスクへと時間を移行することができる。この期間に、時間を寄付したタスクは、それらが与えられようとしていた時間量よりも少ない時間、すなわちそれらの現行の許可よ

り少ない時間に関連するコールバックを受ける。寄付するタスクに与えられる時間は、なおもその資源リストに挙げられた時間であり、もしそのタスクがこの期間に既に稼働を開始して強制排除されてされていれば、寄付行為をしているタスクへの配分を変えることはできない。このため、図4のステップ61及び62に示すように、3Dのようなタスクにとって、もしそれが超過する期間での利用率における非常に早い時期に報告することができれば、非常に効果がある。利用率はこの期間の始期に知られていた処理されるポリゴン数の直接のファンクションであるから、これは3Dにとってかなり些細なことである。

#### 【0103】

負荷移行が実行されるか否かは、ポリシー・ボックスによって管理される。負荷の移行は、ユーザーによって無効にされてもよく、あるいは、負荷を移行する新しいタスクのセットを追加してもよい。

#### 【0104】

##### 非周期的タスク(Non-Periodic Tasks)

最も単純な形式のプロセッサ資源分散装置によって期待される周期的な高度な呼出時間モデルに、完全には適合しないタスクもある。これには非周期的タスクや、呼出時間の必要性が非常に低いタスクが含まれる。このいずれかの特徴を持つタスクは、異なる方法で取り扱われる。

#### 【0105】

##### 非周期的タスク(Aperiodic Tasks)

非周期的のタスクは、本来の周期を持たない。休止タスクは、この一例と考えてよい。但し、実施例においては休止タスクを休止状態のときでなく稼働しているときの動態によって特徴づけられている。非周期的の休止タスクもあり得るが、ほとんどの休止タスクは周期的である。非周期的モデルは、タスク作成及び他の予告できない家政タスクの支援に使用することができる。

#### 【0106】

非周期的タスクは、散発的なサーバーによって取り扱われる。散発的なサーバーは、作業に待ち行列をチェックしながら非常に少ない利用率で周期的に稼働する。作業が可能な場合、サーバーは許可をさらに増やすよう資源マネージャー5

に連絡して要求する。周期的なサーバーのための資源リストへの最小エントリーは、図4のステップ50に示すように、散発的なサーバーがアイドル状態であるが、なすべき作業のチェックをしているモードに使用される。作業が到来すると、同じ最小のエントリーを有するが利用率の増大を要求するエントリーも有する新しい資源リストが、資源マネージャーに提示される。装置が過負荷になると、散発的なサーバーに許可が与えられ、サーバーはその許可を非同期的な作業をするために使用する。作業が完了し、待ち行列が空になると、散発的なサーバーは、資源リストに最小なエントリーだけをして資源マネージャーを再度呼び出す。

#### 【0107】

散発的なサーバーのもとで稼働中のタスクのために実現される最良の呼出時間は、新しい許可を実施するのに要する時間量のファンクションである。最悪の成功例においては、これはすべての未使用の許可期間の合計である。唯一の非同期タスクが装置内にある場合でも、最悪の例は、許容制御が新しい資源リストへのエントリーを拒否することである。これは、装置が過負荷になり、かつすべての稼働中のタスクが散発的なサーバーより優位にされることをポリシー・ボックスが表示した場合にのみ生じる。そのようなポリシーは、ユーザーにより修正されることができる。

#### 【0108】

##### 低呼出時間及び高周波タスク

プロセッサ資源分散装置が支援できる最小呼出時間の一部は、文脈切り換えに要する時間で決定される。それはまた、過剰データ通信量を避けるべく、強制排除と強制排除の間で稼働している通常のタスクの最小単位時間によって、間接的に制限される。（例えば、データはタスクの1つの位相でしばしばアクセスされるが別の位相ではアクセスされないようなキャッシュの中にあってもよい。理想的な文脈切り換え時間は、この2つの位相の間にある。）。

#### 【0109】

プロセッサ資源分散装置は、およそ500マイクロ秒の粒度を提供する。しかし、毎秒1000回（または2000回より多い場合さえある）より多い回数で稼働する必要のあるタスクもあり、それにはこの機構は作動しない。明らかに、

毎秒1000回より多く稼働するタスクは、反復毎に1ミリ秒より少なく使用している。これらは低呼出時間、高周波タスクである。一例は、オーディオ・コマンド・ストリームの代わりに発生するDMA割込みであり、毎秒2800回又は8000回も割り込むことがあるが、各反復に対して少しの命令しか実行されない。さらに、これらの事象に応答する呼出時間要求は、ほんの1マイクロ秒程度である。

#### 【0110】

これらのタスクは、文脈切り換えの時間自体が装置を過負荷にすることが度々あるので、中断ハンドラとして稼働される。中断ハンドラは、完全な文脈切り換えを行わず、中断レベルで利用できる記録器セットも同様に制限される。スケジュール作成のために、プロセッサの幾分かの量が中断処理に保留されている。この量を、どのタスク・セットが稼働されるかによって増加したり減少したりする。これは、プロセッサの100%使用という当社の目標を僅かに損なうが、これらのタスクはそれ自体の待ち時間の必要性より少ない利用率を持たなければならないので、全利用率はあまり大きくない。

#### 【0111】

##### 静的及び動的タスク・セット： レガシー・アプリケーション

プロセッサ資源分散装置は、タスクが起動するとき容認制御を行う能力に依存する。これがなければ、稼働が継続できるように、又はどのタスクが首尾良く稼働するかに関するポリシーをユーザーが安心して損なうことができるように保証をすることができない。

#### 【0112】

如何なるタスクも最後にはドライバを介する要求を開始しなければならないので、ウィンドウズの環境において、またセット最上ボックスの環境においてこれが行われている。

#### 【0113】

DOS環境においては、これらの保証をしていない。DOS環境には、ハードウェアの端子対に直接アドレスすることができると期待するレガシー・アプリケーションがある。ドライバ又は他のOS・APIを介して行われるセットアップ

はない。プロセッサは、DOSボックスにおいてプロセッサ・ドライバに予めロードすることにより、これらのアプリケーションを支援する。また、可能な各タスクのために資源リストを予めロードすることができる。容認制御は、操作系ロード時間に静的に効果的に行われる。初期許可セットは、すべての予めロードされたデバイスが稼働していると推測することを予め設定されている。ウィンドウズがDOSのトップにロードされていれば、この配分とスケジュールは、動的に作成され且つその作成とタスク削除を調整できるものと差し替えられることに注目してほしい。

#### 【0114】

##### 許可条件

タスクに許可が与えられるとき、装置は特別な保証を与えられている。プロセッサ利用率の許可のため、装置は許可された量のプロセッサ時間が各指定期間にタスク配分されることを約束する。一例として、3Dグラフィックスは、各14ミリ秒のうち7ミリ秒を配分されてきた。14ミリ秒毎に3Dグラフィックスは7ミリ秒を与えられるであろう。この7ミリ秒は、期間の始まりに来てもよいし、期間の終わりに来てもよい。あるいは、数個の部分に分割されタスクがそれらの間に強制排除されていてもよい。タスクに時間が配分されずに過ぎることのできる最長時間は $(2 * \text{期間} - 2 * \text{利用率})$ である。このタスクに関しては、 $28 - 14 = 14$ ミリ秒である。このタスクは、この例を処理するに十分なバッファリングを有することが必要である。

#### 【0115】

##### 降伏(Yielding)

この装置の周期的なタスクについての推測の1つは、排除要求に応じる以外に、この期間に作業を終了するまでプロセッサに降伏しない（ブロック）ということである。それらタスクは、外部の事象を待つことをブロック（妨害）しない。散発的サーバーの文脈で稼働する非周期的タスクは、いつでもブロックする可能性がある。タスクは、起動されると、散発的サーバーの作業待ち行列に配置され、その待ち行列上に配置されたものとして実行される。

#### 【0116】

非周期的タスクが閉塞すると、この期間に対するその許可の残部を受けることが期待できない。問題は、容認制御の保証が、プロセッサ時間のすべてのミリ秒を装置が使用できると当てにしていることである。1つは3毎に1ミリ秒（33%の率）を、他は10語とに5ミリ秒（50%の率）を許可されてきた。利用率全体で僅かに83%である。

#### 【0117】

ここで、1つの事象を待つ1ミリ秒の後、10ミリ秒のタスクがプロセッサを明け渡し、その事象が7ミリ秒で戻るときに稼働を続けることができるでしょう。3ミリ秒のタスクは、ミリ秒1で、また再びミリ秒4で稼働する。理論的には、一旦事象が戻ると、10ミリ秒のタスクは次の4ミリ秒を7から10まで使用して、その最終期限を満たすことができる。しかし、3ミリ秒の期間を有する第2のタスクも、それらのミリ秒、7か8か9の1つを使用しなければならない。このスケジュール作成は適合できない。

#### 【0118】

ただ1つのタスクで証明される、閉塞に関するより単純な問題がある。そのタスクは、既に過ぎたか、又は残る利用率時間が最終期限までの時間より大きいかのいずれかであるため、最終期限がもはや満たされない場合に起動することができる。

#### 【0119】

##### 用語集

容認制御：新しいタスクが稼働を許されるか否かを決定するプロセス。

#### 【0120】

非周期タスク：自然の期間を持たないタスク。例えば、短期のユーザー・コマンド要求を実行するタスク。

#### 【0121】

較正ループ：どの程度のプロセッサ利用率が必要とされるかを決定するために稼働される資源リストに指定されたファンクションの反復のセット。較正ループの結果は、資源リストの記入に使用できる。

#### 【0122】

コンテキスト・スイッチ：プロセッサで稼働しているタスクが、1つから別のものへ切り替えられる時。

【0123】

EDF：最早最終期限優先：最も早い最終期限を有する稼働待ち行列上のタスクを次に稼働するよう選択するスケジュール作成手段。単純な状況において、このアルゴリズムは、どの方法でもスケジュールされる如何なるタスクのセットをもスケジュールすることが可能であると証明されている。

【0124】

例外：帯域外通知。例外を受けると、タスクPCがその例外ハンドラーに設定される。

【0125】

許可：ある（反復）期間についての時間の割り振り。例えば、30ミリ秒毎に10ミリ秒のプロセッサ。

【0126】

許可セット：装置に許容されたすべてのタスクに対する許可のセット。

【0127】

最小記入：最小の利用率率を有するタスクの資源リストへの記入。

【0128】

最良記入：最大の利用率率を有し、ユーザーに最高質のサービスを提供するタスク資源リストへの記入。

【0129】

始期：周期的なタスクのための次の期間の始まり。

【0130】

終期：現行の許可が満了した時点。

【0131】

ポリシー・ボックス：ユーザーの望むタスクとサービスの命令を報告するファンクション。どのタスクを稼働させるべきか、またでのレベルの質のサービスかを決定するために装置が過負荷になったとき使用される。

【0132】

強制排除：プロセッサが現在稼働中のタスクから取られて別のタスクへ与えられるとき、稼働中であったタスクは強制排除されると言われ、プロセッサからそれを除去する行為を強制排除と言う。

【0133】

サービスの質（QOS）：アプリケーションによって提供される質のレベル。アプリケーションは、負荷を下げるとき、より低い質のサービスを提供する。

【0134】

休止タスク：装置に容認されたがスケジュール作成にはまだ適格でないタスク。休止タスクは、その期間を放棄している。休止を解かれると、新しい許可セットが計算されるが、容認制御は発生する必要がない。

【0135】

資源リスト：所定のタスクにとって意味のある有用な期間と利用率の組み合わせを規定する仕様のセット。過負荷状態のとき、容認制御を行ったり負荷を下げるのに使用される。

【0136】

資源マネージャー（資源管理機）：装置においてタスクに時間を配分するためのコード。手段に関し、ポリシー・ボックスが参照される。資源管理装置は、容認制御にも、許可セット決定にも関与する。

【0137】

スケジューラ：予め規定されたタスクのセットのうちいずれが次に稼働すべきかを選択すること及びプロセッサ利用率許可を実施することに関与する機能。

【0138】

セット最上ボックス：PCではなく、おそらくユーザー・インターフェイスを有し、かつ利用率可能な媒体デバイスを作る非分散型装置。「セット・トップ」とは、テレビジョン・セットの上にあることを指す。

【0139】

散発的サーバー：自ら周期的に稼働するタスクであるが、本来は非周期的であるタスクに実行時間を提供するタスク。

【0140】



タスク：ユーザーによってジョブを行う実行の単一な流れ。例えばMPEG復号に関わるタスクは多数あり得る。それらタスク間の同期化はその周期性によって処理される。

【0141】

降伏：タスクがプロセッサを断念して休止状態に入るとき。

【0142】

上記の実施例は本発明を例示するに過ぎない。これらに代わる多数の実施例が可能である。本発明は、特許請求の範囲によって定義される。

【図面の簡単な説明】

【図1】

プロセッサ資源分配機のスケジューリング構成要素を示すブロック図

【図2】

3つのタスク用プロセッサ資源分配機のスケジューラにより実行されるサンプル・スケジュールを示す図

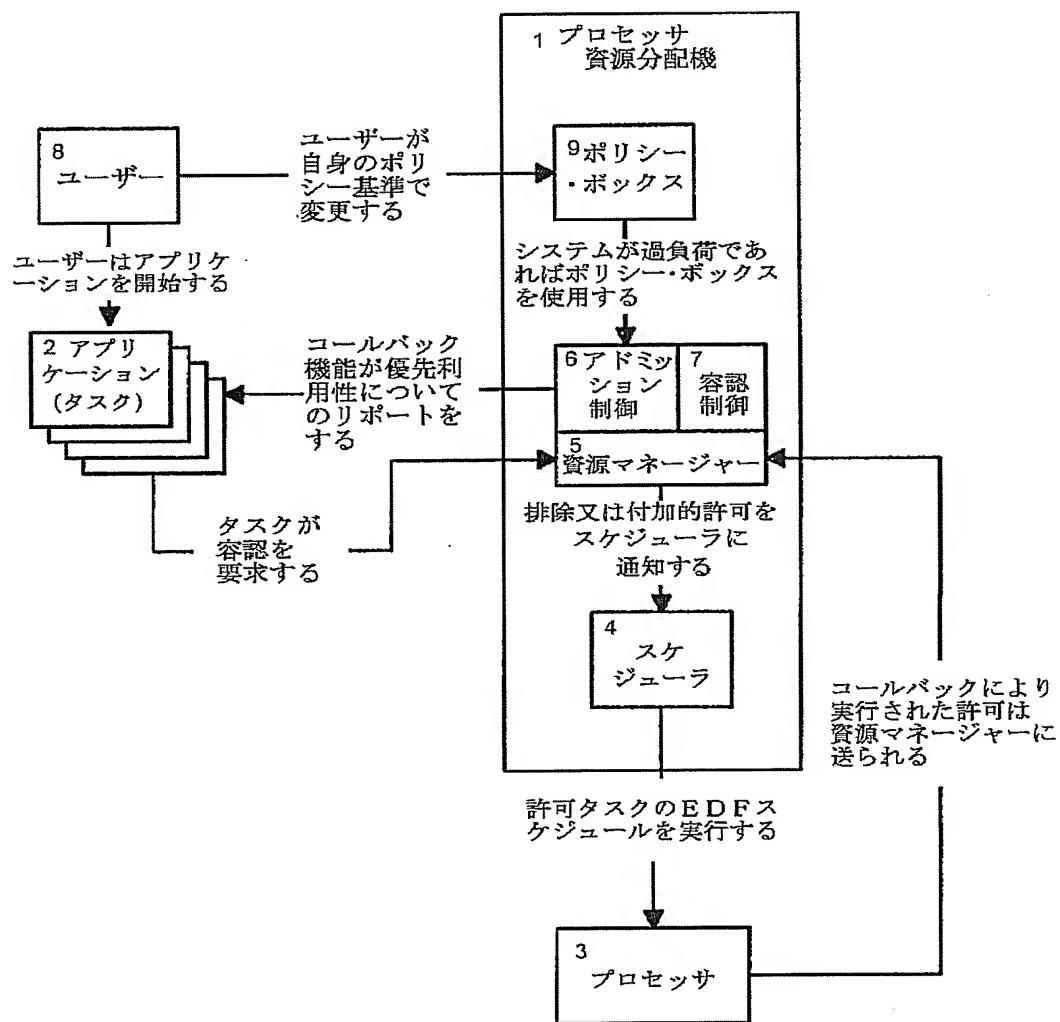
【図3】

資源マネージャの操作を示すフローチャート

【図4】

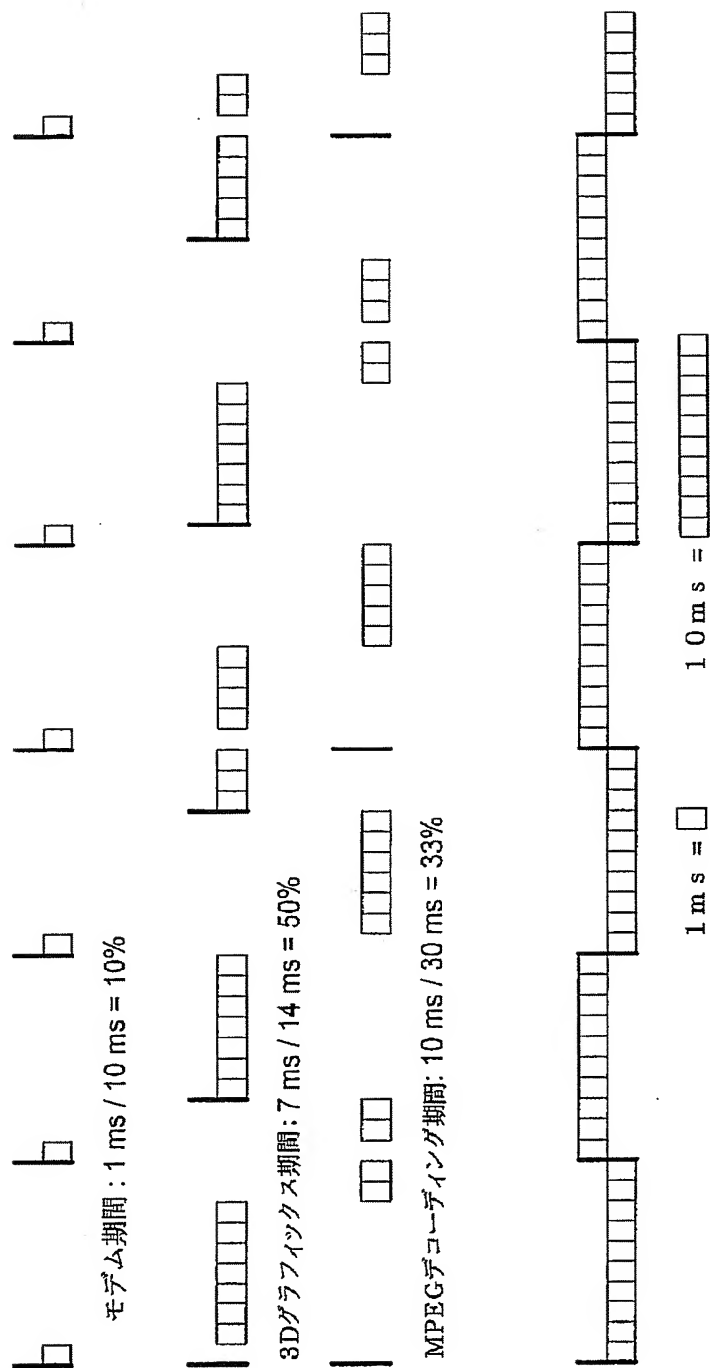
スケジューラの操作を示すフローチャート

【図1】



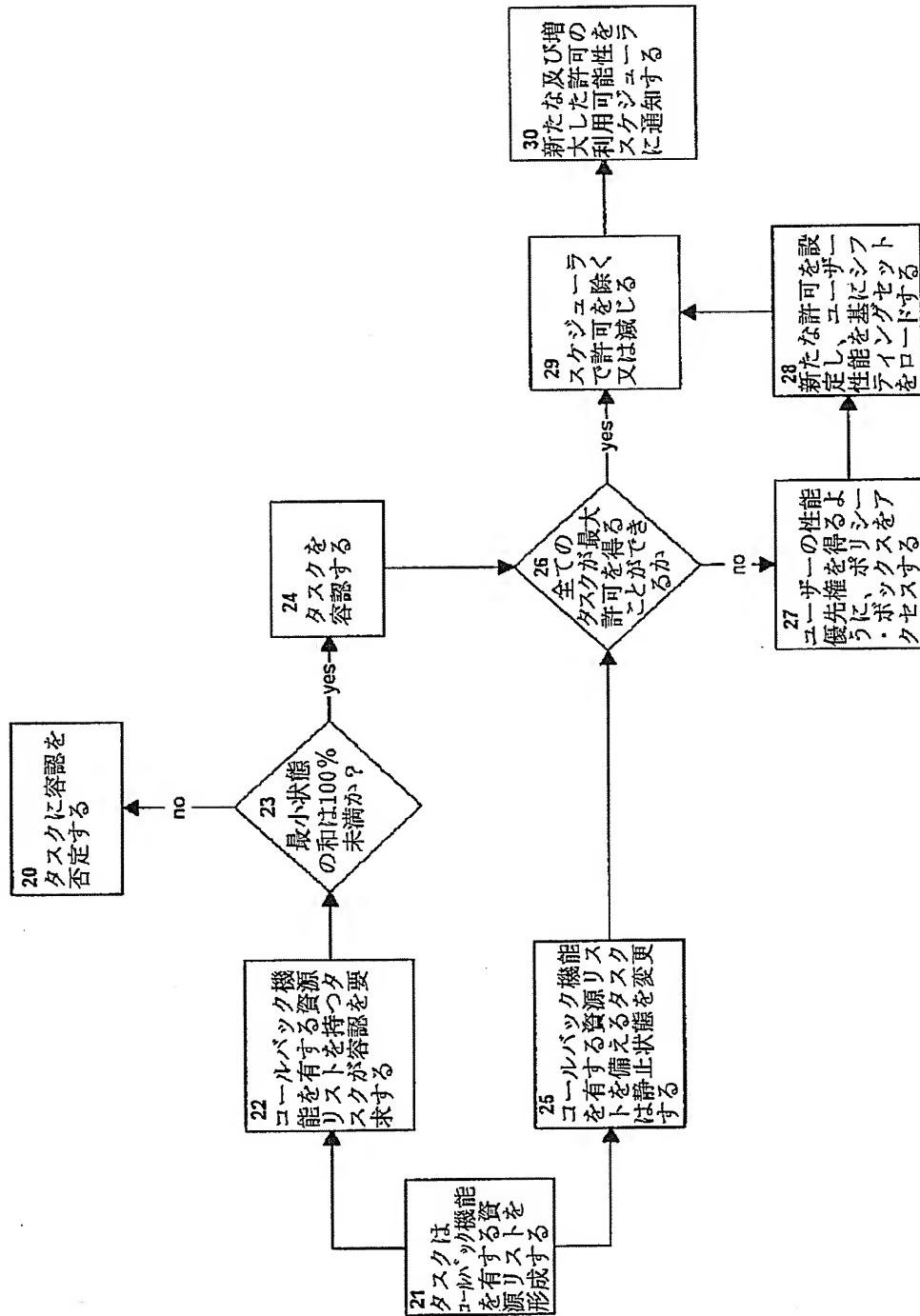
プロセッサ資源分配機のスケジューリング要素

【図2】

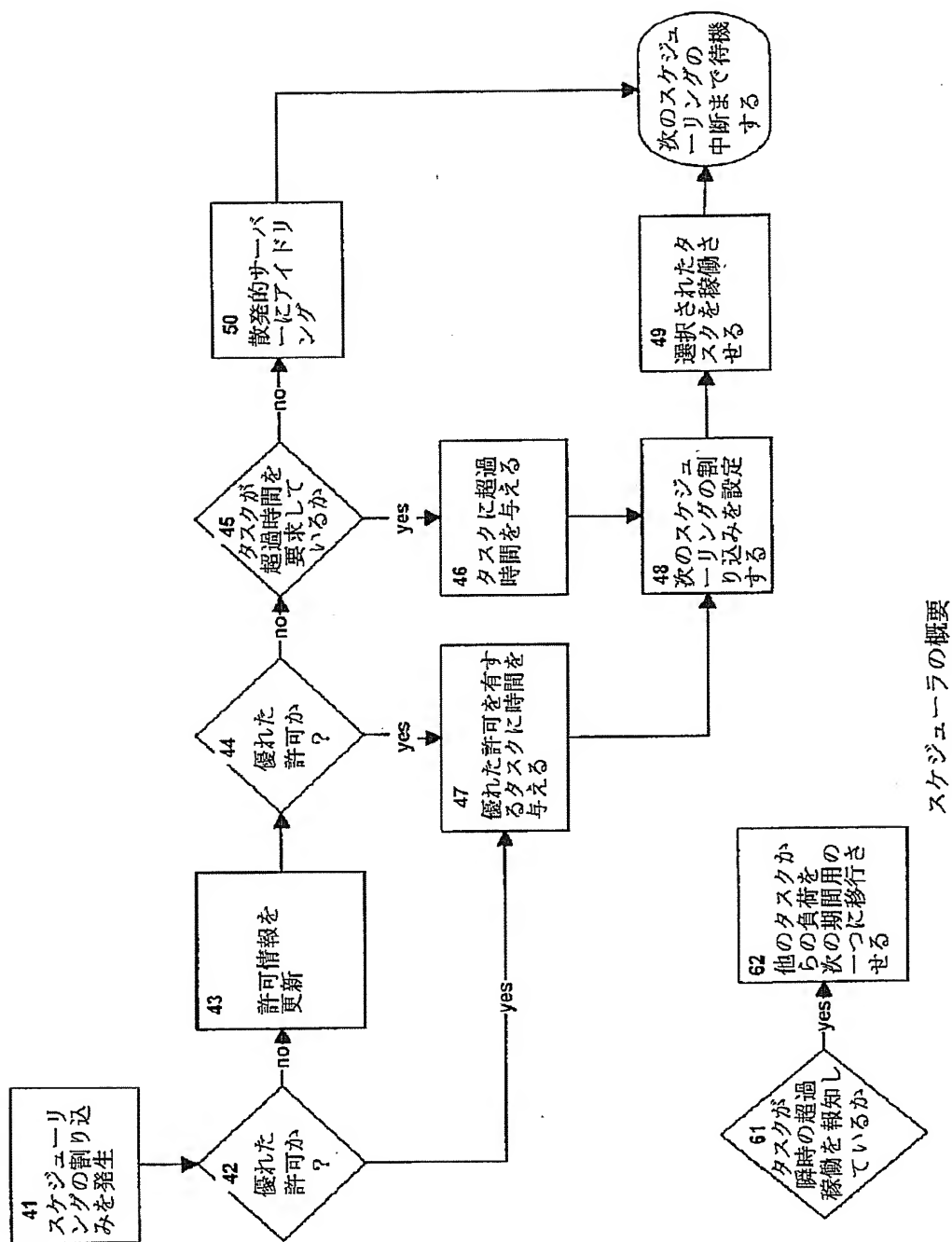


モデム、3Dグラフィックス及びMPEGデコーディングタスク用の66ミリ秒にわたるスケジューリングの時間: 垂直バーは各タスク用の期間の最終期限

【図3】



【図4】



スケジューラの概要

## 【国際調査報告】

## INTERNATIONAL SEARCH REPORT

 International Application No.  
PCT/US 98/18530

 A. CLASSIFICATION OF SUBJECT MATTER  
IPC 6 G06F9/46

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

 Minimum documentation searched (classification system followed by classification symbols)  
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	KANEKO H ET AL: "Integrated scheduling of multimedia and hard real-time tasks" PROCEEDINGS. 17TH IEEE REAL-TIME SYSTEMS SYMPOSIUM (CAT. NO.96CB36024). PROCEEDINGS 17TH IEEE REAL-TIME SYSTEMS SYMPOSIUM, LOS ALAMITOS, CA, USA, 4-6 DEC. 1996, pages 206-217, XP002089217	6,8,9, 13-15, 17,23, 25,26, 30-32, 34,40, 42,43, 47-49,51
A	ISBN 0-8186-7689-2, 1996, Los Alamitos, CA, USA, IEEE Comput. Soc. Press, USA see page 206, right-hand column, line 18 - page 210, left-hand column, line 27  see page 212, right-hand column, line 5 - page 213, left-hand column, line 12 --- -/-	1-5, 18-22, 35-39

☒ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

## \* Special categories of cited documents:

 "A" document defining the general state of the art which is not  
considered to be of particular relevance

 "E" earlier document but published on or after the international  
filing date

 "L" document which may throw doubts on priority claim(s) or  
which is cited to establish the publication date of another  
citation or other special reason (as specified)

 "O" document referring to an oral disclosure, use, exhibition or  
other means

 "P" document published prior to the international filing date but  
later than the priority date claimed

 "T" later document published after the international filing date  
or priority date and not in conflict with the application but  
cited to understand the principle or theory underlying the  
invention

 "X" document of particular relevance; the claimed invention  
cannot be considered novel or cannot be considered to  
involve an inventive step when the document is taken alone

 "Y" document of particular relevance; the claimed invention  
cannot be considered to involve an inventive step when the  
document is combined with one or more other such docu-  
ments, such combination being obvious to a person skilled  
in the art.

"S" document member of the same patent family

Date of the actual completion of the international search

6 January 1999

Date of mailing of the international search report

25/01/1999

Name and mailing address of the ISA

 European Patent Office, P.O. Box 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 240-2040, Tx. 31 651 epo nl,  
Fax (+31-70) 240-3016

Authorized officer

Bijn, K

## INTERNATIONAL SEARCH REPORT

International Application No.  
PCT/US 98/18530

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	NAKAJIMA T ET AL: "A continuous media application supporting dynamic QOS control on Real-Time Mach" PROCEEDINGS ACM MULTIMEDIA '94, PROCEEDINGS OF ACM MULTIMEDIA 94, SAN FRANCISCO, CA, USA, 15-20 OCT. 1994, pages 289-297, XP002089218 ISBN 0-89791-686-7, 1994, New York, NY, USA, ACM, USA	6,8,9, 13-15, 17,23, 25,26, 30-32, 34,40, 42,43, 47-49,51
A	see page 290, left-hand column, line 19 - page 291, left-hand column, line 55  see page 292, right-hand column, line 54 - page 294, left-hand column, line 38 see page 296, left-hand column, line 32 - right-hand column, line 41	1-5, 18-22, 35-39
A	MERCER C W ET AL: "Processor capacity reserves: operating system support for multimedia applications" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS (CAT. NO.94TH0631-2), PROCEEDINGS OF IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS, BOSTON, MA, USA, 15-19 MAY 1994, pages 90-99, XP002089219 ISBN 0-8186-5530-5, 1994, Los Alamitos, CA, USA, IEEE Comput. Soc. Press, USA see page 90, right-hand column, line 5 - page 94, right-hand column, line 13	1-6,8,9, 13-15, 17-23, 25,26, 30-32, 34-40, 42,43, 47-49,51
A	GB 2 304 211 A (FUJITSU LTD) 12 March 1997  see page 27, line 8 - page 30, line 3 see page 38, line 8 - page 46, line 14 see page 51, line 17 - page 55, line 10 see page 57, line 1 - line 14 see page 59, line 22 - page 61, line 4  -/-	1,2,4,6, 9,14, 17-19, 21,23, 26,31, 34-36, 38,40, 43,48,51

## INTERNATIONAL SEARCH REPORT

International Application No.  
PCT/US 98/18530

## C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>EP 0 194 533 A (WANG LABORATORIES) 17 September 1986</p> <p>see column 1, line 35 - line 43 see column 8, line 53 - column 11, line 5; figure 2</p> <p>-----</p>	<p>6, 9, 13-15, 17, 23, 26, 30-32, 34, 40, 43, 47-49, 51</p>



**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International Application No.

PCT/US 98/18530

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
GB 2304211 A	12-03-1997	JP 9054699 A	25-02-1997
EP 0194533 A	17-09-1986	US 4736318 A	05-04-1988
		AU 587427 B	17-08-1989
		AU 5421586 A	04-09-1986
		CA 1248637 A	10-01-1989
		DE 3681609 A	31-10-1991
		JP 2028924 C	19-03-1996
		JP 7056634 B	14-06-1995
		JP 61262938 A	20-11-1986
		US 4908750 A	13-03-1990

## フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW

(72)発明者 カワグチ、 アツオ

アメリカ合衆国 98115 ワシントン州  
シアトル ナンバー ビー 201 ノース  
イースト 95ス ストリート 132

Fターム(参考) 5B098 AA09 FF08 GA04 GA05 GA08  
GC03 GC05 GC10 GD02